KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Acoustic Modeling of Under-resourced Languages

Reza Sahraeian

Supervisor:
Prof. dr. ir. D. Van Compernolle

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

21 November 2017

Acoustic Modeling of Under-resourced Languages

Reza SAHRAEIAN

Examination committee:
Prof. dr. A. Bultheel, chair
Prof. dr. ir. D. Van Compernolle, supervisor
Prof. dr. ir. H. Van hamme
Prof. dr. ir. J. A. K. Suykens
Dr. ir. F. de Wet
(Stellenbosch University, South Africa)
Prof. dr. ir. D. van Leeuwen
(Radboud University Nijmegen, Netherlands)
Prof. dr. ir. C. Wellekens
(EURECOM, France)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

21 November 2017

Preface

This PhD thesis includes the main outcome of my research works in the last 5 years in the realm of speech recognition and processing. Like human beings, an intelligent system may also be learned to write down the spoken words! With this PhD thesis, there are some more approaches available to facilitate this learning process in impoverished scenarios, and I would like to take this opportunity to thank all the people who directly or indirectly contributed to this thesis and making my PhD journey an amazing period of my life.

First and foremost, I would like to thank my supervisor Prof. Dirk Van Compernolle for giving me the opportunity to pursue my PhD under his supervision. I had a challenging time at the beginning of my PhD to attain the objectives as the research area was to a large extent different from my master and bachelor background; however, I always found Dirk patient, understanding and supportive for which I owe him a great deal of gratitude. I also appreciate his trust and the freedom he gave me to find my research path as well as his guidance and the fruitful discussions I had with him without which I would not have been able to accomplish my PhD.

I would also like to express my gratitude towards Prof. Hugo Van hamme and Prof. Johan A. K. Suykens whose helps and advices during my PhD have been truly helpful. Many thanks to Prof. David van Leeuwen and Prof. Christian Wellekens for accepting to be a part this journey and their remarks and advices to make this thesis more informative. Also many thanks to Prof. dr. A. Bultheel the chair of the examination comittee. Special thanks go to Dr. Febe de Wet not only for being a part of jury but also for her helps and advices from a very early stage of my PhD as well as being a wonderful host when I visited CSIR center in South Africa.

I am grateful for the nice colleagues and friends I had during these years at KU Leuven and the memorable time we shared together during the trips, speech group activities, coffee breaks, ping-pong breaks and etc. Prof. Patrick

Wambacq, Annitta, Patricia, Kseniya,Mehmet, Emre, Joris D., Deepak, Hasan, Xueru, Gyorgy, Sayeh, Bart, Kris, Jort, Vincent, Lyan, Jeroen, Alec, Yinan and Wim thank you all. Special thanks to Joris, the longest office mate I ever had; who took his time to do the Dutch translations required for this thesis and also for being such a great friend and colleague. Also I would like to thank Lynn, Ricardo, Giulio, Federica and all other lovely friends with whom I have a lot a of good memories.

During living in Belgium, I was lucky to meet many Iranians who made me feel at home from a very first day of being in Leuven. I would like express my genuine gratitude for all the amazing moments I shared together with Iman, Amin & Baharak, Amirhossein & Neda, Mohammadreza & Neda, Sina & Mahtab, Amin & Bita, Mohsen & Anali, Hasan & Farkhondeh, Hadi, Saeed & Azam, Taha & Maryam, Asefe, Morteza & Zinat, Salim & Armita, Ali & Vida, Ashkan, Mahoor, Neda, Amir, Pooya, Keyvan, Masoud, Ali, Marziyeh, Hamid & Sahar, Sepideh, Mohammad, Mojtaba, Alireza, Mamad, Haleh, Soheil, and also my good old friend Hadi.

I owe a lot to my family in every aspect of my success and I would like to express my gratitude for their unconditional love and support. While being far from a big part of my family, I was truly blessed to have Shima and Behrooz and lovely Fatemeh around and I am so thankful for all the moments we spent together. Moreover, many thanks to my dear brother-in-law Yousef, and all the joys he brought to us during the 3 years of living in Leuven.

While words cannot describe my feelings, I would like to express my heartfelt thanks and love to my wife Zahra. I appreciate all you have done for me. Without a doubt, your love, the positive energy you wrapped me in, your support and encouragement are my main sources of energy and motivation in life, and I would like to deeply thank you for being such an amazing friend and lovely wife.

Abstract

Over the past decades, speech recognition has dramatically improved in a large variety of applications. This improvement can to a large extent be attributed to an increase in computing power and the availability of more speech data. Modern speech technologies require a large amount of speech data to deliver top ASR performances. However, collecting proper data for the ASR task is cumbersome and only for a handful of popular languages we already have abundant amounts of data. Accordingly, to develop a state-of-the-art speech technology for any language the biggest barrier is to collect a lot of data in that language. Thus, developing an ASR system for low resource languages has become an important topic in the community in recent years.

Acoustic modeling is one of the major components of an ASR system which performance highly depends on the amount of training data; thus, this thesis focuses on the challenge of acoustic modeling in low resource settings. We approached this problem from different angles; a big part of our work is in the context of crosslingual and multilingual ASR systems as they have shown to be the most successful strategies to improve on the ASR for low resource languages; in one chapter of this thesis, we also investigate monolingual low resource ASR systems.

First, we propose to employ phone merging to train a multilingual DNN with a universal output layer; our experiments are conducted on two similar languages: Flemish and Afrikaans. The target language in this setting is Afrikaans with only 1hr of data available for training. and Flemish is the high resource donor language. We examined two knowledge-based and data driven phone mapping techniques; we have shown that both of them outperform the multilingual DNN with language-dependent output layers. We also observed that the data driven method performs slightly better than the knowledge-based one.

Next, we set out to find a speech representation with connections to the articulatory features. Articulatory features are connected to the speech

production mechanisms and have interesting properties. These features provide a more compact representation of the speech and therefore acoustic modeling can be accomplished with fewer parameters. An acoustic model with a small number of parameters is less reliant on the training data size and this can be helpful in a low resource setting. Furthermore, articulatory features are more universal and language independent than conventional spectral features and it can be beneficial for crosslingual and multilingual ASR systems. In this work, we utilize Intrinsic Spectral Analysis (ISA) manifold learning as a feature transformation to obtain articulatory-like feature. We conduct several experiments in monolingual, crosslingual and multilingual settings and demonstrated the usefulness of ISA.

Next, we focus on improving on the adaptation of multilingual DNNs to a low resource language. The idea is to reduce the number of updatable parameters in the multilingual DNN so that with a small amount of data, adaptation can still succeed. To this end, we utilize DNN compression with low rank factorization. We show in a set of experiments that by properly compressing a huge multilingual DNN, the performance is improved specifically during the adaptation to a low resource target language.

Finally, we aim at improving on multilingual DNNs by taking an elegant training approach which takes language similarities into account so that for a given target language more relevant information from source languages can be exploited. We propose two methods: one is based on distributed DNN training and weighted model averaging which allows individual source languages to impact the multilingual DNN differently. The second proposal is provided in the framework of ensemble learning where an ensemble of DNN acoustic models derived from different source languages together with the one derived from the conventional multilingual DNN is first constructed; then, through a learning process with the cross-entropy objective function, the constituents are combined via a weighted averaging of the DNN linear components.

Beknopte samenvatting

Automatische spraakherkenning is de voorbije decennia drastisch verbeterd voor een groot aantal applicaties. Deze verbetering is grotendeels te wijten aan verhoogde rekencapaciteit en de beschikbaarheid van meer spraakdata. Moderne spraaktechnologieen vereisen een grote hoeveelheid spraakdata om topprestaties af te leveren. Het verzamelen van geschikte data voor spraakherkenning neemt echter veel tijd in beslag en enkel voor een handvol populaire talen hebben we reeds voldoende data ter beschikking. De grootste hinderpaal bij het bouwen van een krachtig spraakherkenningssysteem voor eender welke taal is dan ook het verzamelen van data. Het is daarom dat het ontwikkelen van een spraakherkenningssysteem voor talen met beperkte middelen een belangrijk onderwerp is geworden in de wetenschappelijke gemeenschap.

Akoestische modellering is een van de hoofdcomponenten van een spraakherkenningssysteem waarvan de performantie sterk afhangt van de hoeveelheid trainingsmateriaal; daarom focust dit proefschrift op de uitdaging van akoestische modellering met beperkte middelen. We hebben dit probleem benaderd vanuit verschillende invalshoeken: een groot deel van ons werk bevindt zich in de context van taaloverschrijdende en meertalige spraakherkenningssystemen, omdat deze zich hebben bewezen als de meest succesvolle strategieën om spraakherkenning te verbeteren voor talen met beperkte middelen; in een hoofdstuk van dit proefschrift onderzoeken we ook eentalige spraakherkenningssystemen met beperkte middelen.

Eerst stellen we voor fonemen samen te smelten om een meertalig DNN te trainen met een universele uitvoerlaag. We experimenteren daarbij met twee gelijkaardige talen: Vlaams en Afrikaans. De doeltaal bij deze experimenten is Afrikaans waarvan we maar 1u data beschikbaar hebben om te trainen. Vlaams is dan de donortaal waarvan we veel data beschikbaar hebben. We hebben zowel een kennisgebaseerde als een datagebaseerde techniek gebruikt om fonemen op elkaar te mappen en hebben aangetoond dat beide het beter deden dan het meertalige DNN met taalafhankelijke uitvoerlagen. We hebben ook ondervonden

dat de datagebaseerde methode het iets beter doet dan de kennisgebaseerde methode.

Vervolgens zijn we op zoek gegaan naar een spraakrepresentatie die verband houdt met de uitspraakkenmerken. Uitspraakkenmerken zijn gerelateerd aan het spraakproductiemechanisme en hebben interessante eigenschappen. Deze kenmerken bieden namelijk een compactere representatie van de spraak waardoor akoestische modellering verwezenlijkt kan worden met minder parameters. Een akoestisch model met een klein aantal parameters is minder afhankelijk van de hoeveelheid trainingsdata en dit kan helpen als men slechts over beperkte middelen beschikt. Uitspraakkenmerken zijn ook universeler en taalonafhankelijker dan conventionele spectrale kenmerken en kunnen gunstig zijn voor taaloverschrijdende en meertalige spraakherkenningssystemen. In dit proefschrift gebruiken we *Intrinsic Spectral Analysis* (ISA) *manifold learning* als een kenmerktransformatie om uitspraakachtige kenmerken te bekomen. We voeren verschillende eentalige, meertalige en taaloverschrijdenden experimenten uit en tonen het nut van ISA aan.

Daarna focussen we op het verbeterd aanpassen van meertalige DNNs naar een taal met beperkte middelen. De idee is om het aantal aan te passen parameters in het meertalige DNN te verkleinen zodat aanpassing nog steeds succesvol is voor een beperkte hoeveelheid data. Om dit te verwezenlijken gebruiken we DNN compressie met lage rang factorisatie. We tonen in een reeks experimenten aan dat het zorgvuldig comprimeren van een gigantisch meertalig DNN de performantie verbetert tijdens de aanpassing naar een taal met beperkte middelen.

Tenslotte beogen we het verbeteren van meertalige DNNs door een elegante trainingsaanpak die rekening houdt met gelijkaardige talen zodat voor een gegeven taal relevantere informatie van verschillende brontalen kan gebruikt worden. We stellen twee methodes voor: de eerste methode is gebaseerd op gedistribueerde DNN training en gewogen modelgemiddeldes wat individuele brontalen toelaat om het meertalige DNN anders te beïnvloeden. De tweede methode situeert zich in het raamwerk van ensemble learning waarbij eerst een ensemble van DNN akoestische modellen dat afgeleid werd van verschillende brontalen wordt gebouwd, tezamen met een model dat werd afgeleid van het conventionele meertalige DNN. Vervolgens worden de constituenten aan de hand van een leerproces met de kruisentropie doelfunctie gecombineerd via een gewogen gemiddelde van de DNN lineaire componenten.

Abbreviations

ASR	Automatic Speech Recognition
CE	Cross Entropy
DD	Data Driven
DNN	Deep Neural Network
EM	Expectation Maximization
FBANK	filter-bank
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IPA	International Phonetic Alphabet
ISA	Intrinsic Spectral Analysis
KB	Knowledge-based
KLD	Kullback-Leibler Distance
LB	Linear Bottleneck
LDA	Linear Discriminant Analysis
LE	Laplacian Eigenmaps
LM	Language Model
LRF	Low Rank Factorization
MFCC	Mel Frequency Cepstral Coefficient

MIDA Mutual Information Discriminant Analysis
MLLT Maximum Likelihood Linear Transform
MLP Multi-layer Perceptron

NCHLT National Center of Human Language Technology

PDF Probability Density Function
PER Phone Error Rate
PLP Perceptual Linear Prediction

RBF Radial Basis Function
RKHS Reproducing Kernel Hilbert Space
ReLU Rectified Linear Unit

SGD Stochastic Gradient Descent
SGMM Subspace Gaussian Mixture Model
SVD Singular Value Decomposition

tanh Hyperbolic Tangent

UBM Universal Background Model

VTLN Vocal Tract Length Normalization

WER Word Error Rate

List of Symbols

$\mathbf{A}$	Matrix containing all intrinsic vectors
$\mathbf{a}$	Intrinsic vector containing intrinsic coordinates
a	Intrinsic coordinate
a	Transition probability between two states in HMM
$\mathcal{A}$	Transformation matrix in DNN linear component
b	Observation probability in HMM
$\mathcal{B}$	Bias term in DNN linear component
c	Mixture weight in GMM
$\mathbf{D}$	Graph degree matrix
D	Dimensionality of data in original space
d	Dimensionality of data in mapped space
E	Quadratic Renyi Entropy
$\mathbf{f}$	Matrix of mapped data with function f
f	Function learned based on manifold learning to map data
G	Group size for g-maxout network
$\mathbf{h}$	Vector of neurons' output
$\mathcal{H}_K$	Reproducing kernel Hilbert space
$\mathbf{I}$	Identity matrix
$\mathcal{I}$	Number of Gaussian components
$\mathbf{K}$	Kernel matrix
K	Kernel function
K	Number of context-dependent sates in HMM/DNN system
$\mathcal{K}$	Kernel matrix used for Parzen estimate of PDF
κ	Kernel function used for Parzen estimate of PDF
$\mathbf{L}$	Laplacian matrix
L	Number of hidden layers in DNN
M	Subspace matrix to calculate means in SGMM
$\mathcal{M}$	Manifold

n_H	Number of neurons in DNN hidden layers
$\mathcal{N}$	Gaussian density function
$\mathbf{P}$	Probability
$\mathcal{Q}$	String of sub-word units
$\mathbf{S}$	Speech signal
S	Number of source languages
s	State in HMM
$\mathcal{S}$	Set of data points
$\mathtt{v}$	State specific vector in SGMM
$\mathbf{W}$	Affinity matrix
w	Affinity matrix element
$\mathcal{W}$	Word string
$\mathtt{w}$	Word
$\mathtt{w}$	Subspace matrix to calculate weights in SGMM
$\mathbf{X}$	Set of feature vectors
x	Speech frame
y	Label assigned to x
$\mathbf{z}$	Pre-nonlinearity of layers in DNN
$\mathbf{1}_m$	Vector of m ones
$\boldsymbol{\alpha}$	Vector of posterior probabilities
α	Posterior probability
$\mathbf{\Gamma}$	Diagonal matrix of eigenvalues
ϵ	Neighbourhood area
θ	Linear component in DNN (including $\mathcal{A}$ and $\mathcal{B}$)
Θ	Set of θ's
λ	Combination weights for combining DNNs
Λ	Set of λ's
μ	Mean of Gaussian
ρ	Parzen window size
Σ	Covariance matrix of Gaussian
σ	RBF kernel hyperparameter
τ	Gaussian similarity function hyperparameter
ϕ	Nonlinear function in neural net
ξ	Manifold regularization parameter
$\mathbf{\Psi}$	All parameters in HMM/GMM

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic Speech Recognition (ASR) is an important part of a speech-based interface between machine and human; such a system mainly aims at converting audio to text. The utility of ASR system has become more pronounced with the emergence of smart and wearable mobile devices; in addition, for different humanitarian, economical and political purposes, companies and governments are interested in leveraging ASR systems. Nowadays, ASR systems are ubiquitous and play different roles in human life such as: personal assistance in Siri, Google home and Alexa, speech translation, voice control of apparatus, automatic subtitling of a video. In most of the applications, ASR systems are used together with other intelligent systems like machine translation or understanding.

A typical ASR system is built based on three sources of information: *language model*, *lexicon* and *acoustic model*. A language model is a statistical model which is trained on text; it contains the probability of the sequence of the words and it expresses how likely one word may come in a combination of other words. The lexicon contains a list of the words with their sub-word phonetic mapping. Preparing the lexicon usually requires expert linguists; however, for a large vocabulary system with millions of words creating the lexicon with some automatic method like grapheme-to-phoneme (G2P) conversion is a more practical approach. Acoustic models are usually considered as the core of speech recognizers and they are learned statistically to model acoustic units of spoken language as they map the continuous speech signals to a sequence of linguistic symbols (phonemes or other symbol units). Acoustic modeling entails estimating a large number of parameters to model sub-word units introduced in the lexicon; this is a supervised process which often requires large amounts of transcribed

speech data. Several decades of research has brought an adequate level of maturity for ASR systems by improving these three major components [67]; the improvement is mainly due to more successful statistical modeling schemes as well as the availability of more data and powerful computing machines.

However, speech recognition can still be a very challenging task. The dynamic properties of speech is the biggest barrier to extracting and modeling reliable patterns for the speech units. This dynamic behavior originates from different sources such as context dependency, noise and environment effects, reverberation, speaker variability, accent, and channel distortion. To overcome this problem, one such approach is to provide large amounts of data to cover all possible variabilities while the system is being learned so that in the test phase the system will not be surprised with a lot of new patterns. Therefore, the training data sparsity can be a big hurdle to deliver top ASR performance. The focus of this thesis is on such a scenario that not a lot of training data is available for a specific language which is called a *low resource* or *under-resourced* language.

1.1 Impact of Training Data Size on ASR Systems

Speech recognition has a long history and in spite of many progresses made, it is still a challenging task. Six major challenges for the speech recognition technology is presented in [67] where the first one is the amount of available training data. The importance of training data size for learning algorithms has long been known and [3] even argued that having more data is more important than a better algorithm. Nowadays we have the opportunity to collect a massive amount of data thanks to the Internet and ever-present devices with voice systems, and thus it is possible to follow the Mercer's famous comment: "There is no data like more data".

The first ASR systems were mainly trained on English; The earliest data sets used for English ASR systems like TIDIGITS and Resource Management include only up to a few hours of data and have a small vocabulary size. Later on, Wall Street Journal (WSJ) corpus emerged with around 80 hours of training data and quite a large vocabulary, and later in 2005, the Fisher data set with around 2000 hours of data was released. Due to the huge positive impact of the training data size, the process of data collection has been continuing and for example Google has very recently reported ASR experiments on 15000 hour voice search task [157].

Figure 1.1, originally presented in [67], shows a good overview of the performance of an English ASR system and how the modern speech recognition systems benefit from more training data. However, we should note that proper data

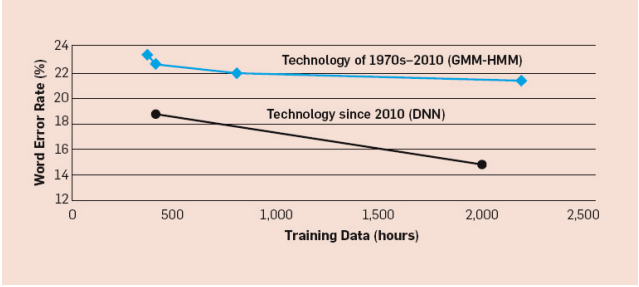


Figure 1.1: Word Recognition Error Rate vs training data size for an English ASR system [67].

collection for ASR systems is cumbersome and for non-English tasks the amount of training data can be much smaller [21].

1.2 ASR for Low Resource Languages

There is no specific definition for low resource languages. In general, when the lack of language resources degrades the performance, it can be deemed a low resource scenario. This can be either because the language resources are really small or the used ASR methodologies demand massive amounts of data. The rest of this section explains the problems and common possible solutions in a low resource setting.

1.2.1 Problem Statement

The low resource problem can be viewed from different perspectives:

- There are about 7000 languages in the world and not all of them are prevalent; some languages are at the risk of extinction and for many of them there are not many native speakers [87]. Accordingly, lack of linguistic experts and human resources makes it difficult to prepare lexicon and transcribed speech data for acoustic model training. Also, not for all languages enough written material can be found to train a reliable language model.
- For some languages it is possible to collect language resources; however, creating the language resources is very costly and time consuming.

Specifically collecting transcribed speech for acoustic modeling is cumbersome; not only does it require human resources, but also it is a very slow task; it can take up to 9 minutes to transcribe one minute of spontaneous speech [5]. Hence, to have an ASR system rapidly and at a reasonable cost, it is of interest to come up with approaches that can deliver a decent performance with a small amount of data.

- As explained in Section 1.1, the more data we have, the better performance we get. This implies that even if it was possible and we did collect a reasonable amount of data, we still can get a higher performance if we prepare more data; this makes the low resource definition relative. Again, because of the cumbersome data collection process, investigating ASR systems which can learn from data collected in other languages is of great value as they are less reliant on the amount of language specific training data.

Among the required sources of data for an ASR, transcribed speech recordings are the most cumbersome to collect; since these data are mainly used for acoustic modeling, a major bottleneck for low resource languages is to train reliable acoustic models which is the focus of this thesis.

1.2.2 Methodologies

Data collection is the ideal solution to deal with low resource scenarios. However, due to the data collection difficulties, several other approaches have been explored to improve acoustic modeling for low resource languages; they generally fall into two categories: monolingual and multilingual.

- Monolingual approaches generally aim at providing robust acoustic models which can be trained with fewer parameters and thus a small amount of training data is required. For example, a common solution is to tie or share the model parameters. The problem can also be tackled in the front-end where a better and simpler feature distribution makes acoustic modeling less complex and consequently modeling can proceed with smaller amounts of data.
- By far, the most successful approach to tackle the low resource problem is to exploit out-of-language data. To this end, data are taken from other high resource (or even low resource) languages; in this thesis, these auxiliary languages are called *source languages* and the language of target is referred to as *target language*. The core idea of multilingual and crosslingual acoustic modeling is parameter sharing; that is, source language(s) are

used to train a set of parameters in the model and only the remaining parameters need to be trained with low resource target language data. Another popular term in this area is *transfer learning* which refers to transferring the knowledge learned in one or more tasks to another target task [105]. This knowledge is usually transferred by porting a part of model parameters across tasks; for our purposes, a "task" may be a language.

1.3 Objectives

The goal of this thesis is to improve acoustic modeling for low resource languages. The most successful strategy to boost acoustic modeling for a low resource language is to rely on multilingual and crosslingual acoustic modeling. Accordingly, we mainly focus on this class of techniques and investigate some related issues to further improve the performance. The multilingual acoustic modeling scheme we predominantly use is based on Deep Neural Networks (DNNs) as they have shown a great promise in speech technology. We look for approaches towards reducing language dependence of DNNs and also providing a better framework to utilize a big multilingual DNN in a low resource setting. Towards these goals, we examine a feature space analysis to obtain less language dependent features. We also aim at improving the adaptation performance which can be burdened by the huge size of DNNs. In addition, to further benefit from specific source languages, we look for approaches that leverage similarities between the source and target languages.

We also investigate a feature space analysis in low resource monolingual acoustic modeling. Inspired by the fact that a more compact feature representation can be modeled with fewer parameters, we investigate the role of proper feature engineering in low resource settings.

Moreover, it is very important to address how sensitive the approaches are to the amount of training data and the identity of the languages. In this respect, we usually provide various settings with different languages and with different amounts of training data in our experiments.

1.4 Main Contributions

The main contributions of this thesis is summarized below:

1. Investigating knowledge-based and data driven phone mappings for the similar languages: Flemish and Afrikaans. The multilingual DNN trained with the proposed phone mappings outperforms the multilingual DNN with language dependent output layer.

2. Employing Intrinsic Spectral Analysis (ISA) as a manifold learning scheme to improve on a low resource monolingual ASR system. The improvement is mostly pronounced in very low resource settings with GMM based acoustic modeling.

3. Improving on multilingual and crosslingual DNN training by reducing the language dependence of the speech features. Manifold learning is utilized to extract an articulatory-like representation which is more universal.

4. Improving on multilingual DNN adaptation by compressing the network via Low Rank Factorization (LRF). We propose an efficient framework for multilingual DNN training by utilizing LRF which always outperforms the traditional multilingual DNN.

5. Development of frameworks for multilingual DNN training to exploit language similarities and complementary information from an ensemble of DNN acoustic models. The results show that the proposed multilingual system can be a proper alternative to the conventional multilingual DNN systems.

The performance of the aforementioned works depends on the settings and the amount of available data for the low resource target languages.

1.5 Thesis Structure

This thesis is structured as follows:

- **Chapter 2: Background**

  This chapter briefly reviews an ASR system, Hidden Markov acoustic modeling, DNNs, adaptation and multilingual speech recognition systems; it defines the common terms and introduces the databases used in this thesis.

- **Chapter 3: Phone Mapping for the Flemish-Afrikaans Setting**

  This chapter includes sets of experiments where Flemish plays the role of high resource donor language and Afrikaans is the low resource target

language. While multilingual DNNs can be structured so that each language keeps its own phonetic inventory, an alternative is to train it with a universal phone set. The phone merging may be helpful if the languages are similar and the target language is very low-resourced. We investigate multilingual DNN training schemes with different output layers. We provide two knowledge-based and data driven phone mappings to create a universal output layer for multilingual DNN training. The automatic data driven method is based on Kullback Leibler Divergence (KLD), and we show that it always performs better than or equal to the knowledge-based phone mapping.

This chapter is mainly based on the following publication ([126]):

- Reza Sahraeian, Dirk Van Compernolle and Febe de Wet. Using generalized maxout networks and phoneme mapping for low resource ASR-a case study on Flemish-Afrikaans. In proceedings of Pattern Recognition Association of South Africa, pages 112-117, Port Elizabeth, South Africa., Nov. 2015.

- **Chapter 4: Speech Manifold for Monolingual Low Resource ASR**

  To overcome the problem of acoustic modeling for low resource languages, one approach could be in the feature space. Conventional acoustic modeling involves estimating many parameters to effectively model the feature distributions and this is mainly due to the non-Gaussian distribution of typical spectral-based features. Finding a feature space in which feature distributions can be modeled with fewer parameters is the main intent of this chapter. Towards this goal, we propose to use a nonlinear feature transformation based on the speech manifold called Intrinsic Spectral Analysis (ISA) for under-resourced speech recognition. We show ISA features outperform the conventional features for the GMM-based acoustic modeling in low resource settings. With more data (>5hr) and DNN-based acoustic modelings, however, no consistent improvement can be obtained by ISA.

  This chapter is based on the following publications ([120], [125]):

  - Reza Sahraeian and Dirk Van Compernolle. A study of supervised intrinsic spectral analysis for TIMIT phone classification. In proceedings of ASRU, pages 256–260, Olomouc, Czech, Dec. 2013.
  - Reza Sahraeian, Dirk Van Compernolle and Febe de Wet. Under-resourced speech recognition based on the speech manifold. In proceedings of INTERSPEECH, pages 1255–1259, Dresden, Germany, Sept. 2015.

- **Chapter 5: Speech Manifold for Crosslingual and Multilingual ASR**

  In this chapter, we study the utility of ISA features in crosslingual and multilingual settings. Assuming that speech production parameters are residing on a low dimensional manifold embedded in the acoustic space, we exploit ISA as a manifold learning method to obtain an articulatory-like feature representation. The coordinates in the resultant representation of which some have demonstrable phonological meaning are shown to be highly portable across languages. We propose a proper framework in terms of data selection and graph construction to train the coordinates from multilingual data, which allows for training the coordinate space when we have abundant out-of-language data. The utility of this representation is further demonstrated in a number of speech recognition experiments using DNNs in a variety of crosslingual and multilingual scenarios.

  This chapter is based on the following publication ([123]):

  – Reza Sahraeian and Dirk Van Compernolle. Crosslingual and multilingual speech recognition based on the speech manifold. Accepted to be published at IEEE/ACM Transactions on Audio, Speech, and Language Processing.

- **Chapter 6: Low Rank Multilingual DNNs for Improved Adaptation to a Low Resource Target Language**

  DNNs have shown remarkable performance in multilingual scenarios; however, these models are often too large in size that adaptation to a target language with a relatively small amount of data may not fulfil. In this chapter, we utilize LRF by using Singular Value Decomposition (SVD) to compress multilingual DNNs for better adaptation. We also address two problems associated with the LRF scheme and propose a compellingly simple methodology to overcome them. First, factorizing all layers results in a huge drop in performance and consequently a long recovery process is required which is not practically efficient. Secondly, LRF can be viewed as a regularization by which some noise is added to the weight layers; however, factorizing all layers together equates to adding too much noise which results in a bad performance. To mitigate these problems, we propose to apply LRF sequentially. We demonstrate the positive effect of LRF in different multilingual scenarios. It is shown that a compressed multilingual DNN is a better starting point for the adaptation.

  This chapter is based on the following publications ([121], [124]):

- Reza Sahraeian and Dirk Van Compernolle. A study of rank-constrained multilingual DNNs for low-resource ASR. In proceeding of ICASSP, pages 5420-5424, Shanghai, China., March 2016
- Reza Sahraeian and Dirk Van Compernolle. Exploiting sequential low rank factorization for multilingual DNNs. In proceeding of ICASSP, pages 5025-5029, New Orleans, USA., March 2017

- **Chapter 7: Exploiting Similarities Between Source and Target Languages**

  In this chapter, we study the possible impact of mismatch between auxiliary source languages and the target language. The performance of a multilingual DNN for a specific target language may be affected by the choice of source languages. While choosing donor language(s) similar to the target language is a well-known practice for multilingual and crosslingual acoustic modeling, it creates a quandary as the performance of the multilingual DNN also depends on the amount of training data. In this chapter, two related methods based on model combination are proposed to benefit from language similarities without data or language selection and also from an ensemble of DNN acoustic models. DNN parameters trained on or adapted to different source languages are combined by weighted averaging so that the relevant information is exploited. We also propose to learn the combination weights with the cross-entropy objective function. We show that the combination method improves the performance of the multilingual DNN.

  This chapter is based on the following publications ([122]):

  - Reza Sahraeian and Dirk Van Compernolle. Using weighted model averaging in distributed multilingual DNNs to improve low resource ASR. In Workshop on Spoken Language Technology for Under-resourced Languages (SLTU), pages 152-158, Yogyakarta, Indonesia., May 2016.
  - Reza Sahraeian and Dirk Van Compernolle. Cross-Entropy Training of DNN Ensemble Acoustic Models for Low Resource ASR. To be submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing.

- **Chapter 8: Conclusion**

  We conclude this thesis by highlighting the contributions and addressing the challenges and possible research directions for the future works.

Chapter 2

Background

2.1 Automatic Speech Recognition

The goal of Automatic Speech Recognition (ASR) is to transcribe a speech signal $\mathbf{S}$ into a sequence of the words $\mathcal{W}$ [114]. Like many other intelligent systems, ASR consists of training and test phases. In the training phase, the available data is used to train (template) models for the speech units; then, in the test phase, the best matching sequence of the templates is given for an unknown input speech signal.

Typical ASR system components are depicted in Figure 2.1. The first step is to convert a recorded speech signal $\mathbf{S}$ into a set of time-ordered feature vectors $\mathbf{X}$; this is called feature extraction and should be carried out in the exact same way for both training and test speech signals. Given a set of test speech features $\mathbf{X}_{test}$, the goal of recognition is to *decode* it into the most likely sequence of the words $\mathcal{W}^*$ which is done by maximizing the posterior probability $\mathbf{P}(\mathcal{W}|\mathbf{X}_{test})$

$$\begin{aligned} \mathcal{W}^* &= \underset{\mathcal{W}}{\operatorname{argmax}}\, \mathbf{P}(\mathcal{W}|\mathbf{X}_{test}) \\ &= \underset{\mathcal{W}}{\operatorname{argmax}}\, \mathbf{P}(\mathbf{X}_{test}|\mathcal{W})\mathbf{P}(\mathcal{W}) \end{aligned} \tag{2.1}$$

The expansion in (2.1) is obtained using Bayes' rule and the fact that $\mathbf{P}(\mathbf{X})$ is constant in the maximization. $\mathbf{P}(\mathbf{X}_{test}|\mathcal{W})$ is the likelihood of data and obtained from the acoustic models. However, it is not practical to train acoustic models for all possible words; instead, each word is represented by sub-word units and acoustic models are trained for those units. Thus, given $\mathcal{Q}$ as the sub-word expansion of the word string $\mathcal{W}$, what acoustic model provides is the probability $\mathbf{P}(\mathbf{X}_{test}|\mathcal{Q})$. Accordingly, (2.1) can be written as:

$$\mathcal{W}^* = \underset{\mathcal{W}}{\operatorname{argmax}} \sum_{\mathcal{Q}} \mathbf{P}(\mathbf{X}_{test}|\mathcal{Q})\mathbf{P}(\mathcal{Q}|\mathcal{W})\mathbf{P}(\mathcal{W}) \tag{2.2}$$

$\mathbf{P}(\mathcal{Q}|\mathcal{W})$ is obtained from lexicon which introduces the sub-word expansion of the words and also contains different possible pronunciations of the words. $\mathbf{P}(\mathcal{W})$ is encoded by the language model; it is trained on a large quantity of text and contains the probability of observing the word string $\mathcal{W}$. Figure 2.1 shows that the decoding block uses the knowledge from three sources of information: language model, acoustic models and lexicon; this can be explained based on the three probability terms in (2.2). In the rest of this section, the components of an ASR system are described in more detail.

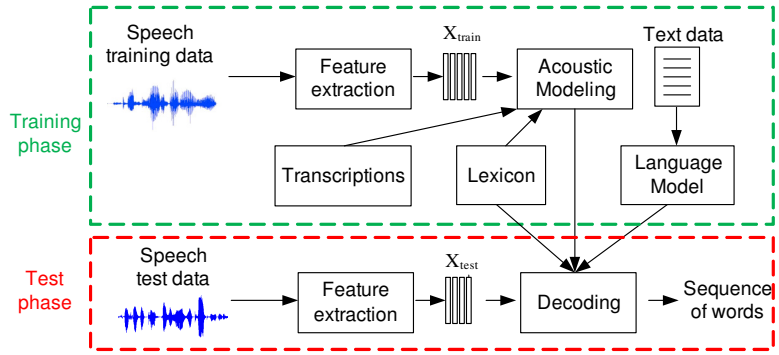


Figure 2.1: An overview of a typical ASR system.

2.1.1 Feature Extraction

The first step in the development of an ASR system typically consists in the computation of low dimensional feature vectors from short overlapping segments of speech. These features usually describe spectral characteristics such as the component frequencies found in the acoustic input and their energy levels. Towards this goal, assuming that over the small window of time speech signal is stationary, the short-time analysis is employed, and for each slice of speech, also called frame, discrete Fourier transform (DFT) [25] is applied to extract a spectral representation of speech. It is a common practice to only take the spectral magnitude and remove the phase information. This power spectrum is then processed through a set of overlapping triangular-shape mel-scale filter banks. The filter banks are motivated by the human auditory system, and they provide compact spectral information for different frequency regions. To compress the range of magnitude, it is also common to apply logarithm on the mel-scale filter banks. The aforementioned preprocessing leads to the well-known mel-filterbank features, which in this thesis, we refer to it as FBANK features.

Another prototypical speech feature type is a compact and decorrelated version of the FBANK features which is called mel-frequency cepstrum coefficients (MFCC) [28]. The MFCC features are obtained by using the so-called cepstral analysis [116]; to this end, the Discrete Cosine Transform (DCT) is applied to the FBANK features and usually only a part of the coefficients, say the first 13 ones, are retained. Another popular cepstral based speech feature is Perceptual Linear Prediction (PLP) [58] which takes a different approach from MFCC to mimic the behavior of human hearing but similarly decorrelates and compresses the FBANK features. It is also common to add the first order (delta) and

second order (delta-delta) derivatives [42] over time to the original spectral features; this is mainly done to compensate for some assumptions made in the acoustic modeling (Section 2.1.2) and also to take the sequential characteristic of speech into account. The correlation between successive feature vectors may also be explicitly accounted for in the emission probabilities themselves [165].

Moreover, feature transformation techniques are amenable to be used in the feature processing. Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two of the most popular linear methods in the speech recognition community. More sophisticated variants of LDA such as mutual information discriminant analysis (MIDA) [33] and heteroscedastic linear discriminant analysis (HLDA) [38] have also shown a great success in this area.

In this thesis, a short-time Fourier analysis is performed with a 25ms Hamming window and a 10ms window shift followed by vocal tract length normalization (VTLN) [175]. VTLN is a speaker normalisation technique based on the fact that different speakers have different vocal tract sizes. For FBANK features, each frame is represented by a 23 dimensional[1] log mel-spectrum applying triangular shaped filterbank using the full spectrum.

Finally, it is worth mentioning that while feature engineering was being a prominent stage in the speech recognition systems for a long time, in the recent years, there have been plenty of attempts to integrate the feature processing with the acoustic modeling step in the framework of DNNs [46, 129, 155]. These methodologies, however, entail complex DNN structure and a large amount of training data and don't show improvement so far.

2.1.2 Hidden Markov Models for Acoustic Modeling

Given the speech features, $\mathbf{X}_{train}$, the common trend in acoustic modeling is to train statistical models for sub-word units presented in the lexicon. Each word string in the training sentences is represented by a sequence of sub-word units $\mathcal{Q}$ and to allow possibility of multiple pronunciations for one word, the likelihood of data is defined as:

$$\mathbf{P}(\mathbf{X}_{train}|\mathcal{W}) = \sum_{\mathcal{Q}} \mathbf{P}(\mathbf{X}_{train}|\mathcal{Q})\mathbf{P}(\mathcal{Q}|\mathcal{W}) \tag{2.3}$$

Acoustic models should be capable of handling sequential and time-varying nature of speech. To this end, the Hidden Markov Models (HMMs) have proven

[1] For a part of the experiments, due to the different default settings, FBANK features have 24 dimensions; this, however, is trivial in our work.

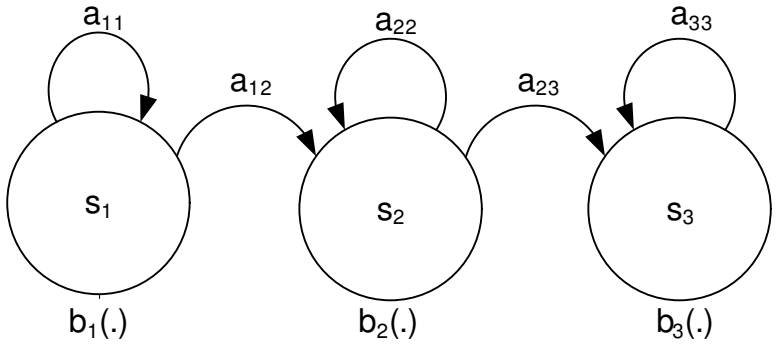


Figure 2.2: 3-state left to right HMM phone model.

successful [115]. In a context-independent (monophone) ASR system each subword phone is modeled by an HMM of a kind shown in Figure 2.2. An HMM consists of emitting states $s_1, .., s_q$; there are two sets of probabilities associated with an HMM: observation and transition probabilities. The observation probability b_j for state j gives the probability of observing x_t at that state. The transition probability a_{ij} is the probability of transition from state i to j. Over the course of HMM training, these two sets of probabilities are estimated so that the likelihood is maximized over the training data. HMM training associates with some assumptions; for one thing, it assumes that observing x_t at state s_j only depends on the previous state which emitted x_{t-1}; this is called first-order Markovian assumption. Moreover, conditional independence assumption which assumes subsequent observations are independent of each other is used to make the computations more tractable. To make up for this inaccurate assumption, we use the popular approach that induces the effect of between frame correlation by using dynamic features like derivatives [42].

To allow different modeling of phones based on their context, context-dependent HMM systems are trained; for example for a triphone context dependent HMM system, each phone is considered based on its left and right neighbouring phones and consequently various HMMs are trained for one central phone. This means that the number of triphones in a context-dependent HMM system can be extremely large and probably because of not having enough training examples for all contexts, the performance will be degraded. To alleviate this problem, state-tying and clustering methods are used [86, 172].

2.1.3 Language Modeling

Language Model (LM) is a prior information for the recognition task. This information is not taken from speech but from the text. Given a sequence of the words $\mathcal{W} = \{\mathtt{w}_1, ..., \mathtt{w}_M\}$, LM is represented as $\mathbf{P}(\mathcal{W})$:

$$\mathbf{P}(\mathcal{W}) = \prod_{i=1}^{M} \mathbf{P}(\mathtt{w}_i | \mathtt{w}_{i-1}, ..., \mathtt{w}_{i-N+1}) \tag{2.4}$$

$\mathbf{P}(\mathtt{w}_i | \mathtt{w}_{i-1}, ..., \mathtt{w}_{i-N+1})$ is the N-gram probability for word $\mathtt{w}_i$; N usually takes the value in the range of 2-5. The N-gram probability for a word $\mathtt{w}_i$ is calculated by counting the number of times $\mathtt{w}_i$ occurs together with the desired $N-1$ other words. For example, in a 3-gram language model, $\mathbf{P}(\mathtt{w}_i | \mathtt{w}_{i-1}, \mathtt{w}_{i-2})$ is estimated by the fraction in which the numerator is the number of times the sequence of $\mathtt{w}_i \mathtt{w}_{i-1} \mathtt{w}_{i-2}$ is observed in the training text corpus, and the denominator is the number of times $\mathtt{w}_{i-1} \mathtt{w}_{i-2}$ is observed.

Noting that recently, deep Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM) units have shown improved performance in LM [77], albeit being computationally expensive. These sequence training techniques are capable of considering a long history of the words into account.

2.1.4 Decoding HMMs

With acoustic models, language model and lexicon, the recognition is a search task expressed in (2.1). One aspect of the evaluation is the acoustic match score obtained by matching the input speech features against the acoustic models. In this thesis, the acoustic models are HMMs trained for sub-word units. These HMMs are further decomposed into states; thus, the decoding is actually finding the optimal state sequences. Finding all possible state sequences is computationally expensive; to make this process efficiently, the Viterbi algorithm is employed [159]. Moreover, the language model provides the probabilities of the word sequences to find the most likely sequence of the words corresponds to the test speech signal. Noting that in this thesis Kaldi ASR toolkit [111] is used for the decoding which is based on weighted finite-state transducers [98].

To measure the accuracy of ASR systems, a standard metric is to simply count the number of mistakes made in $\mathcal{W}^*$. In other words, how many deletions, insertions and substitutions are required to get to the real word sequence from the estimated one in (2.1). The error is termed as Word Error Rate (WER)

and it is normally presented in percentage:

$$WER[\%] = \frac{\#insertions + \#deletions + \#substitutions}{\#words} \times 100 \tag{2.5}$$

Noting that for a part of the experiments in this thesis, the goal is to find the optimal sequence of the phones; hence, the measure of accuracy is Phone Error Rate (PER) defined in a similar way as in (2.5); it is important to note that for the PER recognition experiments we have access to the phone level transcriptions that are being used as references.

2.2 Emission Probability for an HMM

As mentioned earlier, each HMM state is characterized by transition and emission (or observation) probabilities. The emission probability can be obtained in different ways and in this section we describe the popular ones used in this thesis.

2.2.1 Gaussian Mixture Models

A very common way to parametrize the observation probability on each HMM state is by using a mixture of uni-modal Gaussians which is called Gaussian Mixture Model (GMM). The ASR system is correspondingly called HMM/GMM system. More formally, given an observation x_t, the likelihood for state j is:

$$b_j(x_t) = \sum_{i=1}^{\mathcal{I}} c_{ij} \mathcal{N}(x_t; \mu_{ij}, \Sigma_{ij}) \tag{2.6}$$

$b_j(x_t)$ is the probability of observing x_t on state j using the mixture of $\mathcal{I}$ Gaussian components where c_{ij} is the mixture weight with the constraint $\sum_{i=1}^{\mathcal{I}} c_{ij} = 1$. μ_{ij} represents the mean and Σ_{ij} is the covariance matrix. $\mathcal{N}(x_t; \mu_{ij}, \Sigma_{ij})$ is a standard multivariate Gaussian density function given by:

$$\mathcal{N}(x_t; \mu_{ij}, \Sigma_{ij}) = \frac{1}{(2\pi)^{D/2} |\Sigma_{ij}|^{1/2}} \exp\left(-\frac{(x_t - \mu_{ij})' \Sigma_{ij}^{-1} (x_t - \mu_{ij})}{2} \right) \tag{2.7}$$

Where D is the dimensionality of x_t. Training an HMM/GMM system equates to estimating the GMM parameters, i.e. means, covariances and mixture

weights. The training process aims at maximizing the likelihood of training data given these parameters which is also called maximum likelihood estimation [90]. Assuming all the parameters are encapsulated in a set $\mathbf{\Psi}$:

$$\mathbf{\Psi}^* = \underset{\mathbf{\Psi}}{\operatorname{argmax}}\, \mathbf{P}(\mathbf{X}_{train}|\mathcal{Q}, \mathbf{\Psi}) \tag{2.8}$$

$\mathbf{\Psi}^*$ refers to the set of parameters which results in a model that fits the training data. Finding the optimal set of parameters could be easy if we had the state alignment of the training data. What we have in practice, however, is only the transcriptions at the word level. To overcome this problem, a common practice is to employ an axillary cost function through the Expectation Maximization (EM) algorithm [9, 32]. This algorithm estimates the parameters in two steps; in the first step (E-step), with some initial values for the model parameters we pretend to know the state sequence of the training data and we find the statistics required for the second step (M-step). The M-step updates the model parameters. This process continues iteratively until the improvement in the objective is less than a threshold value. Noting that the maximization in (2.8) is not a convex optimization and we may end up with a local optimum.

2.2.2 Subspace Gaussian Mixture Models

The idea of Subspace Gaussian Mixture Models (SGMMs) [109] is to reduce the number of parameters by choosing the Gaussians from a subspace spanned by a background model. Unlike the conventional HMM/GMM systems, the parameters are not directly estimated from the training data, but via a low dimensional model space by capturing the correlation among the context-dependent states and speaker variability. We know that a GMM is characterized by three sets of parameters: means, mixture weights, and covariance matrices. In SGMM modeling, the states have a common structure and share the same covariance matrix, i.e. $\Sigma_{i,j} = \Sigma_i$. The means and mixture weights, however, are allowed to vary along state specific vectors and global mappings. The means and mixture weights in SGMM technique are obtained by:

$$\mu_{ij} = \mathsf{M}_i \mathsf{v}_j \tag{2.9}$$

$$c_{ij} = \frac{\exp \mathsf{w}_i^T \mathsf{v}_j}{\sum_l^{\mathcal{I}} \exp \mathsf{w}_l^T \mathsf{v}_j} \tag{2.10}$$

v_j is the state specific vector; w_i, M_i and Σ_i are globally shared parameters which do not depend on the state. Σ_i is the ith covariance matrix; w_i and M_i span the model sub-spaces for Gaussian means and weights respectively. To train these parameters, a Universal Background Model (UBM) which is a

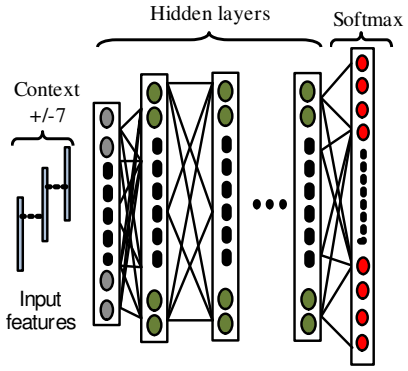


Figure 2.3: A feedforward DNN structure.

mixture of $\mathcal{I}$ Gaussian components is trained to initialize SGMM states and then the parameters are updated using the EM process. Finally, it is important to note that usually (2.9) is extended to include a speaker dependent offset. More details about SGMM training are available in literature [92, 110].

2.2.3 Deep Neural Networks

Acoustic modeling using neural networks has a long history [10]; however, only after the rise of DNNs the acoustic modeling based on DNNs have become the mainstream in ASR [27, 61]. DNN is simply a multilayer perceptron neural network with many hidden layers as shown in Figure 2.3. Training a deep multilayer perceptron is not easy and suffers from multiple problems [49], and only with elegant methodologies for training and initialization [8, 63], the availability of large training data and also more powerful hardwares like GPU DNNs perform very well.

Multilayer perception (MLP) consists of a sequence of linear and nonlinear transformations applied on the input. In this thesis, the nonlinear one is represented by ϕ and the linear component is shown by θ which includes a linear transformation ($\mathcal{A}$) and a bias term ($\mathcal{B}$). DNN is an MLP with multiple layers where at the lth hidden layer:

$$\mathbf{z}^l = \mathcal{A}^l \mathbf{h}^l + \mathcal{B} \quad and \quad \mathbf{h}^l = \phi(\mathbf{z}^{l-1}) \tag{2.11}$$

where the vector $\mathbf{z}^l$ corresponds to the pre-nonlinearity activation and $\mathbf{h}^l$ is the neuron vector. Moreover, whenever we refer to the number of hidden layers

in a DNN, we mean the number of nonlinear hidden layers. For example a DNN with L hidden layers includes $L+1$ linear components ($\{\theta^0, ..., \theta^L\}$) and L nonlinear hidden layers.

The Output layer is a classifier; for the HMM/DNN system, the output layer predicts the posterior probability of each context-dependent state determined by standard clustering algorithms from a previously trained HMM/GMM system. The so-called softmax layer is employed for the purpose of classification:

$$softmax(z^L_{y_i}) = \frac{exp(z^L_{y_i})}{\sum_{\mathtt{k}=1}^{\mathtt{K}} exp(z^L_{\mathtt{k}})} \tag{2.12}$$

$\mathtt{K}$ in the total number of states in the system. $z^L_{\mathtt{k}}$ refers to the $\mathtt{k}$th value of vector $\mathbf{z}^L$, and y_i is the corresponding state label for the frame x_i based on the alignment obtained by the HMM/GMM system.

For a large part of the experiments in this thesis, we used HMM/DNNs for acoustic modeling; therefore, the rest of this section explains some terms and characteristics of the DNNs.

Input features

The input to the DNN is the speech features after short-time signal processing (e.g. FBANKs, MFCCs); usually multiple frames are spliced at the DNN input to contain information from longer temporal context; In this thesis, features are always spliced together with 7 left and 7 right neighbor frames at the DNN input. The HMM/DNN systems are not as sensitive as HMM/GMM systems to the input feature dimensionality as it only affects the input layer. Moreover, it has been shown that unlike GMM models, DNNs can benefit from correlation between the features [97]. Thus, in general there is no need to go from FBANKs to MFCCs and in this thesis we always use the FBANK feature as the HMM/DNN input.

Objective function

The most widely used objective function for HMM/DNN systems is Cross Entropy (CE). CE is measured between the target probability and the output of the softmax. The target probability is taken from the Viterbi alignment which is actually a hard label, i.e. the probability is either 0 or 1. Therefore, the objective function is reduced to:

$$-\sum_{t}\log \mathbf{P}(y_t|x_t) \tag{2.13}$$

y_t refers to the state assigned to x_t. $\mathbf{P}(y_t|x_t)$ is calculated from the softmax output as shown in (2.12).

DNN training

DNN parameters, θs, are most commonly trained via Stochastic Gradient Descent (SGD) [61] where the training data is split into small subsets, called mini-batches. In each pass, the DNN parameters are trained over one mini-batch and throughout one epoch the entire training data is processed. It is also very important for a good estimation that the data be randomly shuffled prior to the training. The common SGD from random initializations performs poorly with DNNs and the success of DNN training only obtained with new initializations and training algorithms. The impact of initialization can be described as a regularizer that set the parameters in a better place in the optimization procedure; in this regard, more analyses can be found in [8, 40]. The initialization can be done in an unsupervised manner with Restricted Boltzmann Machine (RBM) [63] or with supervision using greedy layer-wise training [8]. The DNN training in this thesis follows the latter paradigm: first, a randomly initialized network with one hidden layer is trained for a small number of iterations; then, the weights that go to the softmax layer are removed and a new hidden layer and two sets of randomly initialized weights are added. The neural network is trained again for the predefined number of iterations before the new hidden layer is inserted. This is repeated until we reach a desired number of layers.

Furthermore, for a faster convergence and better performance the choice of *learning rate* is crucial [137]. Learning rate weights the gradients to control the pace of updating the parameters. It has been shown that the best strategy is to schedule learning rate to vary during the training; in the beginning of the training it is large, and it decreases later on when finer moves are required in the optimization space. In this thesis, the learning rate is decreased exponentially by a factor of 10 during training.

Nonlinear activation

To ensure that the neural network is not just an stack of linear layers, having a nonlinear activation function ϕ is important. Different nonlinear functions

have been used for neural networks; Table 2.1 includes the list of the activation functions we use in this thesis:

Table 2.1: The list of some nonlinear activation functions used in DNNs

Hyperbolic Tangent (tanh)	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	1 in, 1 out
Rectified Linear Unit (ReLU)	$\phi(z) = max(z, 0)$	1 in, 1 out
Maxout	$\phi(\{z_i\}_{i=1}^{G}) = max(\{z_i\}_{i=1}^{G})$	G in, 1 out
Generalized maxout (g-maxout)	$\phi(\{z_i\}_{i=1}^{G}) = (\sum_{i=1}^{G} \lvert z_i \rvert^p)^{\frac{1}{p}}$	G in, 1 out

Maxout and g-maxout nonlinearities are applied on a group of nodes with the size G; thus, they can be viewed as dimensionality reduction too [51, 177]. The generalized maxout network calculates the p-norm of the inputs and accordingly it is also called p-norm activation; in this thesis we use $p = 2$.

Finally, it is worth mentioning that nowadays more sophisticated structures of DNNs than the feedforward DNN shown in Figure 2.3 are being the state-of-the-art in acoustic modeling. These types include Convolutional Neural Networks (CNNs) [1], Recurrent Neural Networks (RNNs) [52], Long-Short Term Memory (LSTM) [130] and some combination of them [128].

2.3 Adaptation

Any mismatch between the training and test data degrades the ASR performance. This mismatch arises from different sources such as speaker variability, accent, environment and even languages. A common solution to reduce this mismatch is to specialize the already trained ASR system for the new test scenario; this can be accomplished by *adapting* acoustic model parameters towards the new test case. Adaptation has long been used in ASR systems and it has been employed in a variety of tasks [43, 85, 91, 180]. The success of adaptation highly depends on the amount of adaptation data and the number of parameters that need to be updated with that data.

In the context of DNN-based ASR systems, adaptation is more challenging as the DNN is characterized with a huge number of parameters. In the literature, some studies have shown that not all these parameters need to be updated; for example in [145], only the hidden unit activations are updated, or [170] proposed to restructure the DNN parameters to adapt only a small portion of them; in general adaptation of DNN models is a popular topic in the community. In this thesis, we employ adaptation to shift parameters of a DNN towards the low resource target language.

2.4 Multilingual ASR

Due to the globalization in the last decade, the need for multilingual speech technology has emerged. Hence, multilingual ASR has moved from an academic research topic to an industrial one. In general, if any of the major components in an ASR system is obtained from multilingual sources, the ASR system is called multilingual. The utility of multilingual ASR is not limited to low resource languages. For example, the language model can be trained on text from multiple languages to handle switches between languages (code-switching) [164, 166]. Moreover, multilingual information in the lexicon is helpful for the code-switching scenario as well as accented speech or non-native speech. In this thesis, however, the focus is on acoustic modeling for low resource languages and accordingly the language models and lexicons are kept monolingual.

A large part of this thesis centres around multilingual and crosslingual acoustic modeling. Employing multilingual data to boost acoustic modeling performance for a low resource language or providing language universal components that can be reused in any ASR system has been a research direction for a long time [132]. Multilingual acoustic modeling, however, is not a straightforward task due to the differences like different sets of sub-word units across languages. In the literature, multilingual data is exploited for acoustic modeling mostly in the framework of GMM, SGMM and neural network systems.

2.4.1 GMM based multilingual systems

Early attempts in multilingual acoustic modeling are based on HMM/GMM systems and mainly use direct sharing of phonetic models across languages or use data from related languages to augment the low resource data pool [35, 50, 79]. The effectiveness of language independent speech recognition systems using a universal phone set and language dependent phone mappings was examined by Schultz and colleagues [134, 135]. The universal phone set is constructed by pooling all phoneme units of different languages; this, however, leads to a huge number of acoustic units and is problematic especially in a context-dependent system. To keep the size of the multilingual phone set limited, phone mapping in a knowledge-based [84] and data driven [139] manners are investigated across languages. Moreover, [140] proposed to use universal speech attributes instead of a universal phone set; these attributes characterize speech based on the phonetic features like voicing, nasality and etc. and can be used for any language. Furthermore, by adding a language question while building context-dependent models, data pooling is done in a data driven fashion for contexts in which languages are similar to each other [23].

Despite the advances gained from the aforementioned studies, it was usually found that performance drops quickly when moving from a language dependent scenario to a language independent one unless the amount of training data is really small [156]. Besides, using a universal phone set imposes a constraint on the ASR systems to use lexicons with the same phone units; not only does this bring extra efforts but also acoustic phonetics of each language are best expressed with a specific phone set of that particular language. This degrades the acoustic model performance specifically when languages come from different families.

Later on, parameter sharing between acoustic models proceeded in a more sophisticated way so that each language can maintain its own phone set. One of the most successful ones is SGMM [12]. In SGMM acoustic modeling, parameters are factorized into state-specific and globally-shared sets; the global parameters are first trained in a multilingual fashion and then adapted to a target language [93].

Also, acoustic model adaptation techniques have been widely utilized together with data borrowing to improve the HMM/GMM ASR performance in impoverished scenarios [13, 30, 89, 179]. No matter which multilingual acoustic modeling technique is employed, the parameters can be further updated to shift the model in the acoustic space towards the target language; the performance of adaptation depends on the amount of available data and the number of parameters that should be adapted.

2.4.2 DNN based multilingual systems

A popular use of neural networks for multilingual ASR system is based on tandem features [59] where the output of an MLP is employed as a new feature for recognition task. In this regard, MLP can be trained on multilingual data and then play the role of feature extractor for a new target language data [82,148,154]. The multilingual information in the form of posterior features is also successfully exploited in the framework of KL-divergence HMMs [68]. In recent years, DNNs have been used in a large body of research to exploit out-of-language data particularly for under-resourced speech recognition. The key assumption in multilingual DNNs is that the hidden layers can be considered as a universal complex feature transformation and can be shared across languages while the softmax layers are language dependent. This suggests that the hidden layers can be trained simultaneously for different languages to benefit from each other [47, 65]. Figure 2.4 shows the process of employing a multilingual DNN

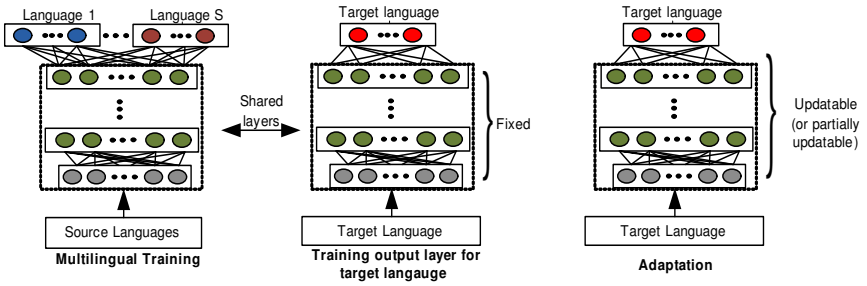


Figure 2.4: Multilingual DNN training with shared hidden layers (left plot). Reusing the hidden layers for a new target language (middle plot), and adaptation of the networks with target language data (the right plot).

for a low resource target language [2]; after the multilingual DNN is trained, the hidden layers are reused and only the output layer is trained for the target language. Furthermore, it has been shown that additional gain over the purely multilingual DNN can be obtained by adapting the multilingual DNN using data from the target language [55]; adaptation usually refers to retraining of the multilingual DNN with the data of target language; this retraining process can be applied only on a part of the hidden layer parameters.

While following the same assumption, multilingual and crosslingual DNNs are exploited in different ways; in the rest of this section, we explain about some of the most frequent ones.

Universal Softmax

A common approach in any multilingual ASR is the creation of a universal phone set by first pooling the phone sets of different languages together and then merging them based on their similarity in both knowledge-based and data driven fashions [84, 135]. The same approach can be taken in the realm of multilingual neural networks by joining of language-specific phone sets or mapping to a global phone set [39, 131]. Having a common phone set means that the same DNN structure as shown in Figure 2.3 can be used in a multilingual case where the softmax layer is taken from an HMM/GMM system trained multilingually. Then, the hidden layers are frozen and the output layer is trained using the low resource target language.

[2] This figure shows the multilingual DNN training based on multi-task learning as will beshown in Figure 2.5. Having said that any multilingual DNN training could be used in Figure 2.4.

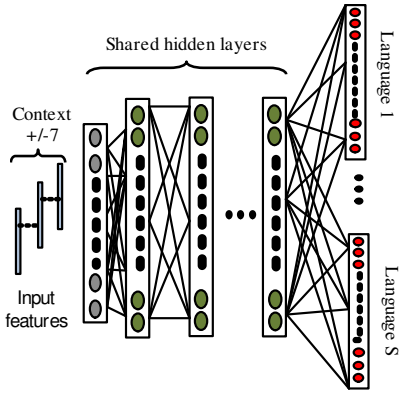


Figure 2.5: Multilingual DNN training with multi-task learning.

Multi-task training

The theory of multi-task learning states that by jointly learning different related tasks which share the same input and some internal representation, the performance of each task can be improved [14]. As shown in Figure 2.5, multilingual DNN training can be accomplished in a multi-task learning framework as the input and hidden layers are shared across multiple languages while each language maintains its own language-specific output layer [65]. For each input speech frame, only the task of the corresponding language is trained; By enforcing common parameters in the hidden layers, the relatedness between the tasks is exploited.

Multi-task learning of DNN has an upside of no need for creating a universal softmax layer although it has been shown that adding such a universal output layer as a separate task can bring further improvement [18]. Moreover, in multi-task learning framework we may control the effect of different tasks on the final learning process; however, for the sake of generalization, usually all the tasks are given the same weight.

Tandem

The hidden layers being trained on multiple languages can be viewed as language-independent feature extractors. The multilingual hidden layers, with an optional bottleneck layer, can be employed to extract features that will serve as input to another GMM or DNN based system [149, 158, 162]. Figure 2.6 shows a DNN with a bottleneck layer. The bottleneck features can be concatenated with

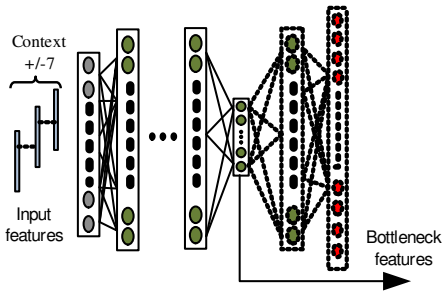


Figure 2.6: Bottleneck feature extractor.

other features like FBANK or MFCCs and also be transformed via conventional feature transformation techniques like LDA. While it is a common practice to take the features from the one before the last hidden layer as shown in Figure 2.6, some researchers have shown that the features from the last hidden layer can also be successfully deployed [171].

2.5 Data sets

In this thesis, we have conducted various monolingual and multilingual experiments using data from different languages.

- **CGN (Flemish)**: The Corpus Gesproken Nederlands (CGN) corpus is a standard Dutch database that includes speech data collected from adults in the Netherlands and Flanders [104]. The corpus consists of 13 components that correspond to different socio-situational settings. In this thesis, only Flemish data from component-O (including read speech) are used. It has 38 hours of speech sampled at 16kHz and we have taken 36hr (150 speakers) for the training and 2hr (8 speakers) for the evaluation. The CGN pronunciation dictionary uses an alphabet of 47 phones.
- **NCHLT (Afrikaans)**: The Afrikaans data is taken from the NCHLT corpus consisting of 210 speakers, including broadband read speech sampled at 16 kHz [4]. The phone set contains 37 phones and silence. All repeated utterances were removed from the original dataset. In our setting, to simulate various low resource conditions, we considered one hour of data, five hours of data and the full training set including about 10.7 hours. We used the default evaluation and development sets including 2.2hr and 1hr data respectively (Table 2.2).

Table 2.2: Used Subsets of NCHLT Afrikaans Corpus.

Set:	Train1	Train2	Train3	Test	Dev
Size	1hr	5hr	10.7hr	2.2hr	1.0hr
# speakers	188	192	192	8	10

Table 2.3: Statistic information of 9 languages from the GlobalPhone dataset. Noting that for some languages diphthongs and triphthongs are also included in Phones.

Language	#Phones	#Speakers	Amount of data(hr)		
			Train	Dev.	Eval.
German (GE)	41	77	14.85	1.95	1.46
French (FR)	38	100	22.74	2.12	2.02
Portuguese (PO)	45	101	22.71	1.64	1.79
Russian (RU)	48	115	21.10	2.68	2.60
Spanish (SP)	40	100	17.55	2.04	1.67
Swedish (SW)	49	98	17.38	2.05	2.19
Turkish (TU)	29	100	13.23	1.96	1.88
Arabic (AR)	44	78	16.54	1.38	1.01
Mandarin (MAN)	139	132	26.6	1.98	2.41

- **GlobalPhone (Multilingual)**: The GlobalPhone corpus is a multilingual text and read speech corpus that covers speech data from 20 languages [133]. Table 2.3 presents the detailed statistics for 9 languages used in this thesis. We carried out sets of classification and recognition experiments in monolingual, crosslingual and multilingual settings. In some scenarios, we only used a subset of data from some languages that we mentioned explicitly in the corresponding experimental sections.

2.6 Settings

In this thesis we have conducted several experiments in different scenarios using the datasets explained in the previous section and the architectures for multilingual systems described in Section 2.4.2. We may categorize the experiments into two groups: 1) *Flemish-Afrikaans* setting where Afrikaans is the target language and Flemish is the source language. 2) *GlobalPhone* setting where up to 6 languages are used as target languages and up to 9 languages play the role of donor languages. While each chapter explains the settings for

the corresponding experiments, in this section we provide an overview of the experimental scenarios being used in each chapter in Table 2.4.

Table 2.4: Overview of the multilingual experiments conducted in the chapters.

	Multilingual setting		Multilingual DNN systems
Chapter 3	Flemish-Afrikaans	✓	Universal softmax
	GlobalPhone	×	-
Chapter 4	Flemish-Afrikaans	×	-
	GlobalPhone	×	-
Chapter 5	Flemish-Afrikaans	×	-
	GlobalPhone	✓	Multi-task learning Tandem
Chapter 6	Flemish-Afrikaans	✓	Universal softmax
	GlobalPhone	✓	Multi-task learning
Chapter 7	Flemish-Afrikaans	×	-
	GlobalPhone	✓	Multi-task learning

It is worth mentioning that in all scenarios we have also presented the monolingual baseline results. These results are presented to show how using a multilingual system improves the performance of a low resource ASR.

Chapter 3

Phone Mapping for Flemish-Afrikaans Setting

This chapter is adapted from the following article(s):

- Reza Sahraeian, Dirk Van Compernolle and Febe de Wet. Using generalized maxout networks and phoneme mapping for low resource ASR-a case study on Flemish-Afrikaans. In proceedings of Pattern Recognition Association of South Africa, pages 112-117, Port Elizabeth, South Africa., Nov. 2015.

3.1 Introduction

In the realm of multilingual neural networks, creating the target phone set for the multilingual training is commonly done along three different approaches (a) by joining of language-specific phone sets, (b) creating a universal phone set by mapping all phones from various languages into a global phone set, (c) and training neural networks where each language has its own output layer as shown in Figure 2.5. While all three scenarios have been investigated there is no consensus on which one is best. This actually depends on the setting; the first two approaches have been successfully used when sufficient amounts of training data for each language is available [158] [131]. When training data is sparse, however, using the information from high resource language(s) by merging phone sets may be beneficial [161].

While multilingual DNNs can be trained as each language has its own output layer, our goal is to investigate if better performance is gained by knowledge-based and data driven phone mappings and which one performs the best for a very low resource target language. The answer highly depends on the languages; for example, if two languages are closely related, an IPA (International Phonetic Alphabet) based phone mapping may work sufficiently well; whereas for the unrelated languages the mapping can be challenging. In this chapter, we conduct a case study for two related languages: Flemish and Afrikaans [56]; after all, phone mapping is more meaningful when there are some similarities among the languages.

The knowledge-based phone mapping is done by utilizing linguistic information and knowledge of the native speakers. The data driven approach we used is based on a confusion matrix by calculating KLD between pairs of the phone distributions in Flemish and Afrikaans. Similar works exist in the literature addressing data driven phone mappings by making confusion matrices using multilingual neural networks [39, 54]. However, the reported performances mostly degrade compared to the knowledge-based method. Moreover, our approach is more flexible as we may assign more than one phone from the source language (Flemish) to each phone of the target language (Afrikaans) based on the confusion scores.

3.2 Phone Mapping in Multilingual DNNs

Creating a universal phone set by simply concatenating language-specific phone sets may degrade the performance since very similar phones from different languages are considered as different classes and the DNN would likely fail

to discriminate between them [158]. Alternatively, a priori knowledge of a phonetician can be used for a knowledge-based mapping which is not always accurate; thus, the DNN may need to encode disparate phones as a single class. This motivates to investigate whether a data driven phone mapping can overcome the aforementioned problems. The rest of this section describes the knowledge-based and data driven phone mappings we used to train the multilingual DNN.

3.2.1 Knowledge-based Phone Mapping

One major assumption for knowledge-based (KB) phone mapping could be the fact that the articulatory representation of the phones are similar and their acoustic realization can be assumed language independent. Based on this idea, universal phone inventories such as the IPA have been proposed [69]. In this work, the pronunciation dictionaries for the Afrikaans and Flemish include 37 and 47 phones respectively. In our KB phone mapping, each phone from the Flemish dictionary is mapped to only one of the phones in the Afrikaans one. To this end, 31 phones that share the same symbol in the IPA table are merged. Table 3.1 shows the list of the phones in both languages using the $*$ mark.

Table 3.1: IPA symbols for Flemish (FL) and Afrikaans (AFR) phone sets

IPA symb.	p	b	t	d	k	g	f	v	s	z	ʃ	ʒ	x	ɣ	h	m	n	əi
FL	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
AFR	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*	*

IPA symb.	ŋ	l	r	j	w	ɪ	ɛ	ɑ	ɔ	ʏ	ə	i	eː	aː	oː	y	u	əu
FL	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
AFR	*	*	*	*	*		*	*	*		*	*				*	*	*

IPA symb.	øː	ɛi	ɑu	œy	ɛː	ɔː	ʏː	ɑ̃	ɛ̃	ɔ̃	ʏ̃	ɦ	œ	æ	ɑː	iə	uə	
FL	*	*	*	*	*	*	*	*	*	*	*							
AFR	*			*								*	*	*	*	*	*	

However, it can be seen that there are 16 phones in Flemish without any IPA counterparts in Afrikaans which are mapped based on the linguistic knowledge. The phones: ɛ̃, ɑ̃, ɔ̃ and ʏ̃ are simply mapped to /ɛ n/, /ɑ n/, /ɔ n/ and /ʏ n/, and the rest are mapped as described in Table 3.2.

Table 3.2: KB phone mapping between Flemish (FL) and Afrikaans (AFR) languages for the phones with no common IPA counterpart.

FL	AFR	FL	AFR	FL	AFR
ɣ	x	ʏː	ə	ɔː	ɔ
h	ɦ	oː	uə	ɛi	əi
ɪ	ɛ	eː	iə	ɛː	ɛ
ʏ	œ	aː	ɑː	ɑu	əu

3.2.2 Data driven Phone Mapping

In our data driven (DD) approach, each phone in Flemish is mapped into the N-best corresponding matches in Afrikaans by measuring the distance between the phones and forming a confusion matrix based on the distances. Afterwards, a new pronunciation dictionary is created in which Flemish entries are described by the Afrikaans phones. Table 3.3 includes two examples explaining how the Flemish words "met" and "stipt" are phonetized in the original Flemish lexicon and the new KB and DD ones. In the first example, the phone "ɛ" in the Flemish is mostly confused with the three phones in Afrikaans: "ə", "œ" and "əi". Therefore, we consider three different pronunciations for this word based on the phone "ɛ" in the new lexicon.

In this setup, the size of the new dictionaries increase rapidly with increasing N values. In addition, many of the Flemish phones have dominant matchings based on the confusion matrix; this is the case for almost all of the consonants. In this study, we set N=1 for the consonants and N=3 for the rest of the Flemish phones. It is also interesting to note that the Flemish phone "ɛ", for example, was merged with the Afrikaans phone of the same IPA symbol in the KB phone mapping. However, "ɛ" is not among any of the three candidates chosen by the DD approach. This indicates how differently the KB and the DD phone mapping may work.

In the second example, three different pronunciations for the word "stipt" are shown based on the phone "ɪ". This phone has no IPA matching in Afrikaans and is mapped to "ɛ" according to the linguistic knowledge as shown in Table 3.2. We should note that although the KB candidate for this phone is among those selected by the DD approach, we have two more possible options for the mapping and depending on the context the best one will be chosen.

Table 3.3: An example of pronunciation modeling using DD and KB phone mapping for Flemish-Afrikaans setting.

FL word	FL lexicon	DD lexicin	KB lexicon
met(1)	m ɛ t	m ə t	m ɛ t
met(2)	-	m œ t	-
met(3)	-	m əi t	-
stipt(1)	s t ɪ p t	s t ɛ p t	s t ɛ p t
stipt(2)	-	s t i p t	-
stipt(3)	-	s t ə p t	-

To generate the confusion matrix, we measure the KLD between the distributions of phones of the two languages. For any phone p in Afrikaans and q in Flemish:

$$KLD(\mathbf{P}^p \parallel \mathbf{Q}^q) = \int \mathbf{P}^p(x) \log \frac{\mathbf{P}^p(x)}{\mathbf{Q}_q(x)} dx \tag{3.1}$$

Where $\mathbf{P}$ and $\mathbf{Q}$ represent density functions of the phone distributions in Afrikaans and Flemish respectively. In other words, (3.1) measures how similar the distribution of phone q in Flemish is to the distribution of phone p in Afrikaans. It is worth noting that since KLD is not symmetric, it is normally appropriate for $\mathbf{P}$ to be the reference distribution and $\mathbf{Q}$ to be an approximation to it [80]. KLD is straightforward for normal distributions. However, for the multivariate GMMs, the KLD is not analytically tractable and therefore we can use the variational approximation of KLD between GMMs [60].

$$KLD^v(\mathbf{P}^p \parallel \mathbf{Q}^q) = \sum_i c_i \log \frac{\sum_{i'} c_{i'} e^{-KLD(P_i \| P_{i'})}}{\sum_j \hat{c}_j e^{-KLD(P_i \| P_j)}} \tag{3.2}$$

Where $\mathbf{P} = \sum_i P_i$ and $P_i = c_i \mathcal{N}(\mu_i, \Sigma_i)$, and $\mathcal{N}(\mu_i, \Sigma_i)$ represents the normal distribution with mean μ_i and covariance Σ_i. Similarly $\mathbf{Q} = \sum_j Q_j$ and $Q_j = \hat{c}_j \mathcal{N}(\mu_j, \Sigma_j)$. c and $\hat{c}$ are the Gaussian weights assigned to the Gaussian mixtures in $\mathbf{P}$ and $\mathbf{Q}$ respectively. KLD^v is calculated for all pairs of the phones in Afrikaans and Flemish to construct the confusion matrix[1]. In this study, the number of Gaussian components is set to 2 empirically.

[1]Noting that for the training data we have access to the phone segmentation files.

Table 3.4: PER(%) using HMM/GMM and HMM/DNN systems with *tanh* activation function trained on 1hr of Afrikaans data.

	HMM/GMM	Hybrid DNN	
		1 layer	2 layers
PER(%)	25.18	24.49	25.35

3.3 Experiments

This section describes the experimental study performed to evaluate the impact of phone mapping on multilingual DNNs where Flemish is the donor language and Afrikaans is the low resource target language. In the first set of experiments, monolingual baseline systems are presented on Afrikaans with 1hr of training data (Train1 from Table 2.2). The second part of our experiments includes multilingual DNN results where Flemish is used as the donor language.

3.3.1 Monolingual Experiments

The first set of experiments was carried out on the Afrikaans language only. We used a standard front-end to extract 13-dimensional features including 12 MFCC coefficients and the energy. Then, first and second derivatives were added and utterance-based mean and variance normalization was applied in both training and test stages. These features were used to build 3-state left to right HMM triphone models with a total number of Gaussian components of $\sim$3000, and the number of context-dependent triphone states was 505.

We trained a bi-gram phone model on the training set and the ASR performance is reported in phone error rate (PER). In this set of experiments, we first trained standard DNN systems with *tanh* activation functions. The input feature was FBANK and the number of units in each layer was 100. Table 3.4 provides the ASR performances using both HMM/GMM and the HMM/DNN systems. Since we have only one hour of training data, increasing the number of hidden layers degrades the performance. The PERs for HMM/DNN systems with 1 and 2 layers are reported in Table 3.4; we observed higher PERs for more hidden layers. The best performance for the monolingual HMM/DNN with *tanh* nonlinearity is obtained with one hidden layer.

Then, we trained the DNNs with the 2-norm activation function; in this case, we have one more hyperparameter to set which is the group size, G. The proper value for G, the input dimensionality for the 2-norm activation and the number of hidden layers were jointly found on the validation set. In Table 3.5,

the PERs for different numbers of hidden layers and different values of G are presented. In these experiments the output dimensionality is 100 and various input dimensionalities are investigated. Table 3.5 shows that the performance is improved when the generalized maxout network is used for such a low resource setting.

Table 3.5: PER(%) for Afrikaans with 1hr of training data using HMM/DNN systems with various settings where the 2-norm output dimensionality is 100.

input dim.	# of hidden layers			
	1	2	3	4
400	23.61	23.83	23.68	23.72
300	23.59	23.96	23.99	24.03
200	23.76	23.71	24.01	24.01

3.3.2 Multilingual Experiments

We subsequently merged the Flemish and Afrikaans training data based on both the KB and the DD phone mappings explained in Section 3.2. Then, we trained a multilingual HMM/GMM system using 39-dimensional MFCC features. The numbers of tied-states used for the multilingual HMM/GMM system were 4131 and 3973 for the KB and DD approaches respectively.

Table 3.6 gives the performances of the multilingual HMM/GMM systems for the two types of phone mapping by using the same bi-gram language model. These results are presented here to evaluate the effectiveness of the DD phone mapping. As shown, DD phone mapping considerably improves the performance of the multilingual HMM/GMM system compared to the KB one; yet, it can be seen that the PER is much higher than the monolingual case presented in Table 3.4 and Table 3.5.

Table 3.6: PER(%) comparison for KB and DD phone mapping using the multilingual HMM/GMM system obtained from Flemish and 1hr of Afrikaans.

	KB mapping	DD mapping
PER(%)	45.89	39.81

Multilingual DNNs were subsequently trained by adopting context dependent decision trees and audio alignments from the multilingual HMM/GMM systems. In this set of experiments, the DNNs used 2-norm activation functions. 2-norm input and output dimensionality were empirically set to 1000 and 200 respectively. To bootstrap the acoustic model for Afrikaans, the hidden layers

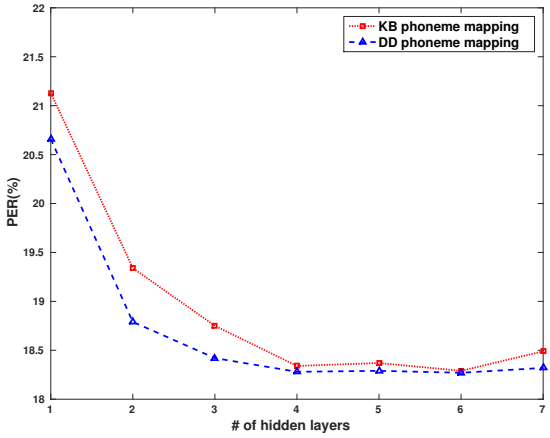


Figure 3.1: PERs(%) comparisons for the KB and DD phone mappings using the multilingual DNN trained on the Flemish-1hr Afrikaans setting.

of the multilingual DNNs were shared and the softmax layer was replaced with the output layer corresponding to Afrikaans.

Figure 3.1 compares the PERs obtained by the multilingual DNNs with different hidden layers and it reveals the following trends: first, both multilingual DNN systems provide significant reductions in PERs compared to the monolingual baseline systems presented in Table 3.4 and Table 3.5. Secondly, a comparison between the KB and DD phone mappings for DNN training shows that the ASR performance is improved by using the DD phone mapping. However, only marginal performance differences are observed if the neural networks are trained deep enough. This difference, however, depends on how similar the results of the two phone mapping techniques are. In this study, we observed that our DD technique maps all consonants to the same Afrikaans phones as the KB mapping does. Moreover, for many of the other Flemish phones, the selected KB candidate is among those chosen by the DD approach. For unrelated languages, however, DD phone mapping may perform differently. Finally, we examined muti-task learning of multilingual DNNs where output targets for Flemish and Afrikaans were kept separate. In this scenario, the hidden layers were trained with data from both languages while the softmax layers were trained with language specific data where the number of output targets were 4113 and 505 for Flemish and Afrikaans respectively.

Table 3.7 shows that multilingual DNN approaches, either with or without phone mapping, improve ASR for the low resource language. Moreover, we

Table 3.7: PER(%) comparison for the multilingual DNNs with 6 layers trained on Flemish and 1hr of Afrikaans with phone mappings (KB and DD) and language dependent output layer.

	Universal target		Language dependent
	KB	DD	targets
PER	18.29	18.25	21.04

observe that the phone mappings considerably improve the performance of multilingual DNNs over the multi-task training with language specific outputs. This can be due to the fact that Afrikaans and Flemish are closely related languages. The performances are reported for the DNNs with 6 hidden layers.

3.4 Conclusion

This chapter presented an investigation of using phone mappings for multilingual DNN based acoustic modelings to improve the speech recognizer for Afrikaans (as an example of a resource-scarce language) by borrowing data from Flemish (as an example of a related well-resourced language). Phone sets of these two languages were merged in both knowledge-based and data driven fashions. We proposed to use an approximation of KLD to generate the confusion matrix for a DD phone mapping. This DD approach led to a performance improvement compared to KB which was more pronounced in the multilingual HMM/GMM system than the HMM/DNN one. Moreover, we observed that if we train neural networks deep enough, the performance of the two phone mapping approaches get closer to each other. We also observed that the phone mapping approaches in our Flemish-Afrikaans setting improves multilingual DNN performance compared to the multi-task learning framework with language specific output layer.

Chapter 4

Speech Manifold for Monolingual Low Resource ASR

This chapter is adapted from the following article(s):

- Reza Sahraeian and Dirk Van Compernolle. A study of supervised intrinsic spectral analysis for TIMIT phone classification. In proceedings of ASRU, pages 256–260, Olomouc, Czech, Dec. 2013.
- Reza Sahraeian, Dirk Van Compernolle and Febe de Wet. Under-resourced speech recognition based on the speech manifold. In proceedings of INTERSPEECH, pages 1255–1259, Dresden, Germany, Sept. 2015.

4.1 Introduction

The main issue which makes speech recognition a challenging task in resource constrained settings is that conventional speech recognizers rely heavily on statistically based modeling schemes and need to estimate a large number of parameters to effectively model speech feature distributions. This is mainly due to the complex distribution of typical features such as PLPs or MFCCs that are usually used. Finding a feature space in which feature distributions can be modeled with fewer parameters is therefore an interesting challenge. This motivates efforts to investigate the impact of feature transformation in the front-end of ASR systems to accommodate low resource language scenarios [150]. The main motivation in this chapter is to use a feature transformation technique which reduces the complexity of the speech feature distribution and improve linear separability.

In this chapter, we investigate the utility of a nonlinear feature transformation in the context of manifold learning for a low resource setting. Manifold learning is a popular framework to learn nonlinear projection maps that recover the underlying configuration space assuming that the high dimensional data resides on or nearby a low dimensional manifold embedded within the high dimensional space. ISOMAP [146], Locally Linear Embedding (LLE) [119], Laplacian Eigenmaps (LE) [6], Diffusion Maps (DM) [100] and manifold regularization [7] are some examples of nonlinear embedding techniques that may drastically reduce the representational dimensionality while preserving the local structure of data points. This class of algorithms has been widely used in machine learning. However, the validity of the manifold structure assumption is necessary for the success of such techniques.

It has been postulated long time ago that the process of generating acoustic signals through the speech production mechanisms defines a nonlinear mapping from articulatory space to the acoustic space [41, 143]. Hence, to recover an articulatory representation of speech, feature transformations can be used to inverse the original nonlinear mapping. The study of feasibility of such nonlinear feature transformations in the context of manifold learning has been first investigated by using a variant of Laplacian eigenmaps named Intrinsic Spectral Analysis (ISA) [72]. ISA has been successfully used in various speech recognition tasks in a supervised, unsupervised and semi-supervised manner [73, 74, 76, 120]. Furthermore, some of these intrinsic features may have a near binary behavior and directly relate to some broad phonetic classes and separate natural classes of speech sounds; linear separability may therefore be easier in the intrinsic subspace [74]. This suggests that acoustic modeling can be accomplished with lower complexity and less data. For example, in GMM based models, fewer Gaussian components are required to reliably train phone models.

Furthermore, to reduce the computational complexity of learning ISA, a data selection is commonly used; for a manifold learning technique, finding a subset of data points which still represents the manifold structure of data is essential. In this respect, we investigate using an entropy-based data selection method.

We conduct sets of recognition experiments for monolingual low resource settings to assess the performance of ISA as a preprocessing for acoustic modeling. The results show that ISA with a subtle data selection is useful for GMM-based acoustic modeling. For DNN-based systems, however, ISA does not bring consistent improvement.

4.2 The Speech Manifold

The manifold assumption seems intuitive for speech signals as they are generated via a human speech production apparatus which has few degrees of freedom and thus cannot produce very high dimensional sounds in acoustic space [41, 151]. In [71], it was proven that the parameters of an acoustic tube model of the vocal tract lie on a manifold embedded in acoustic space and that an inverse mapping must exist that is a coordinate chart on a manifold embedded in the observation space. Given sufficient data points in acoustic space and making modest smoothness assumptions about the manifold, it must therefore be possible to derive the nature of this manifold and to discover the inverse mapping from acoustic to configuration space. In practice we cannot work with the manifold directly, but we may use an adjacency graph to provide a sampled equivalent as is done in graph theory. The assumptions about smoothness and a sufficient numbers of samples may be combined into an assumption that speech data points that are very close to each other in the configuration space will still be measurably close to each other in acoustic space.

There are two issues, however, which are worth mentioning. First, using a simple source-filter model is inadequate to model certain sounds such as obstruents, potentially invalidating the manifold assumption for this class of sounds. Furthermore, as the finite parameter acoustic tube model is approximate at best, it may present another challenge to the manifold assumption or at least imply that some of the learned features may not have a meaningful articulatory interpretation. Yet, as pointed out in [72], the obstruent phonemes still cluster naturally after applying ISA, and as shown in [74], several coordinates in the mapping space have a strong connection to traditional acoustic-phonetic features. These observations give strong support to the underlying manifold assumption, despite the limitations in any practical speech production model that might be used to formally prove its existence.

4.3 Manifold Learning

Manifold learning is a popular approach to nonlinear dimensionality reduction which attempts to reveal the governing parameters assuming that the data points are lying on or close to a low dimensional manifold (intrinsic space) embedded in the representational (extrinsic) space. Several algorithms have been proposed for manifold learning [15]; they generally exploit the local structure of data in the original space and try to preserve it in the new mapped space. The manifold learning method we use in this thesis is ISA and is based on the concepts of Laplacian Eigenmaps (LE) and Manifold Regularization [7]. We review the basics of these techniques in the rest of this section.

4.3.1 Laplacian Eigenmaps

The only criterion used to validate the manifold assumption is locality preservation, i.e. points that are close to each other in the extrinsic space should also be close to each other in the intrinsic space. More formally, we are given a collection of n samples $\{x_1, x_2, ...x_n\}$ that form a mesh of data points that lie on the manifold $\mathcal{M}$ embedded in a D-dimensional space $\mathcal{R}^D$. The goal is to derive a mapping f from $\mathcal{R}^D$ to $\mathcal{R}^d$ with $d \leq D$. The manifold structure is represented by an affinity matrix $\mathbf{W}$ whose elements, w_{ij}, are a measure of the similarity between x_i and x_j. The similarity score is only computed for points x_j that are in a small ϵ-neighborhood of x_i, or vice versa, and is set to zero otherwise as larger distances as measured in extrinsic space are considered to be uninformative with respect to the distance in intrinsic space. In our work, we use the Gaussian similarity function, $w_{ij} = exp(-\|x_i - x_j\|^2/2\tau^2)$, where τ is a hyperparameter. The LE algorithm obtains the desired locality preservation by minimizing the following cost function:

$$\frac{1}{2}\sum_{i,j} w_{ij}\|f(x_i) - f(x_j)\|^2 \tag{4.1}$$

If we denote $\mathbf{f} = [f(x_1)f(x_2)...f(x_n)]^T$ the embedding matrix of size $n \times d$ where $f(x_i)$ is the d-dimensional embedding for x_i, then minimizing (4.1) can be rewritten in matrix format (refer to Appendix A for more details):

$$\mathbf{f}^* = \underset{\mathbf{f}}{\text{argmin}}\ \text{tr}(\mathbf{f}^T\mathbf{L}\mathbf{f}) \tag{4.2}$$

where $\mathbf{L}$ is the graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}$ is the diagonal matrix with elements $\mathtt{d}_{ii} = \sum_{j=1}^{n} w_{ij}$.

After further applying a normalization constraint $\mathbf{f^T D f} = \mathbf{I}$, eliminating the trivial zero-solution and applying Lagrange multipliers, the solution to (4.2) is found by solving the generalized eigenvalue decomposition problem: $\mathbf{L f}^* = \mathbf{\Gamma D f}^*$; where $\mathbf{\Gamma}$ is a diagonal matrix containing the eigenvalues. The final d-dimensional embedding is obtained as the matrix of eigenvectors corresponding to the d lowest (non-trivial) eigenvalues of the above optimization problem. For more details we refer to [6]. Finally note that, for numerical reasons, in practice we employ the normalized version of the graph Laplacian [160]: $\mathbf{L}_{norm} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. This is fully equivalent and further on we will use $\mathbf{L}$ as a shorthand notation for $\mathbf{L}_{norm}$.

4.3.2 Intrinsic Spectral Analysis

The major drawback of the LE framework explained above is that a solution is given for points on the graph and not for the entire manifold; thus, for new data, it requires to recompute everything. To mitigate this problem, ISA has been formulated as an extension to LE using manifold regularization to allow natural extensions to out-of-sample data by seeking for a mapping f in a reproducing kernel Hilbert space (RKHS) to an intrinsic basis on the manifold [7, 72]. This is achieved by solving the following optimization problem:

$$f^* = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \operatorname{tr}(\|f\|_K^2 + \xi \mathbf{f}^T \mathbf{L} \mathbf{f}) \tag{4.3}$$

where $\mathcal{H}_K$ is a RKHS of functions with the corresponding norm $\|\|_K$; K is some positive semi-definite kernel function. The first term in the cost function is the extrinsic norm measured in the ambient space; the second term is a measure for smoothness in the intrinsic space as used in LE above in (4.2) and ξ is a hyperparameter making the balance between both terms. The classical representer theorem states that the solution to this minimization problem for the jth component can be written as

$$f_j^*(x) = \sum_{i=1}^{n} a_i^j K(x_i, x) \tag{4.4}$$

where f is defined as a set of d intrinsic basis functions $\{f_1, f_2, ..., f_d\}$. Then, we need to estimate the new parameter set $\mathbf{a}^j = [a_1^j, ..., a_n^j]^T$ to find the representation of any given new data point on the jth intrinsic coordinate. As explained in Appendix A, by plugging (4.4) into (4.3) and applying Lagrangian multipliers, the following generalized eigenvalue decomposition problem is derived:

$$(\mathbf{I} + \xi \mathbf{L K}) \mathbf{A} = \mathbf{K A \Gamma} \tag{4.5}$$

$\mathbf{K}$ is the $n \times n$ Gram matrix and its ijth element is $K(x_i, x_j)$. We always use a Radial Basis Function (RBF) kernel: $K(x_i, x_j) = exp(-\|x_i - x_j\|^2/2\sigma^2)$. $\mathbf{A} = [\mathbf{a}^1\mathbf{a}^2...\mathbf{a}^n]$ is the matrix containing the eigenvectors and as in the case of the LE the d-dimensional embedding is obtained by only using those eigenvectors corresponding to the d smallest non-trivial eigenvalues that are a solution of (4.5).

There are two main issues associated with all kernel-based methods including ISA in large scale speech applications which are worth being pointed out. The first one is the memory and computational expense arising from the construction of the large affinity matrix of size $n \times n$; moreover, the computational complexity of eigendecomposition scales with the number of samples [142]. Even with a few hours of speech data, we end up with millions of samples. Thus, to mitigate this problem, we need to select a subset of data points usually randomly to train ISA coordinates [72, 74]. The second issue is the many hyperparameters, $(\epsilon, \tau, \xi, \sigma)$, that must be chosen. However, this issue is not severe as it has been claimed [76] that the performance dependency is weak over a large range of values and this is also what we observed in our experiments.

4.4 Data Selection

To handle the first limitation, several approaches have been proposed such as Nyström methods [37, 81] and random Fourier features based methods [66, 117]. For speech applications each sample represents a short segment of speech (e.g. 10msec); hence, even for a very low resource setting of 1 hour of data, the training set includes 360k samples. Therefore, one major drawback of methods like ISA for speech tasks is the construction of the large affinity matrix of size $n \times n$, where n is the number of samples. Moreover, the computational complexity of the eigendecomposition of the Laplacian matrix derived from the affinity matrix scales with the number of samples. Nyström approximation method was used to solve a reduced eigenvalue problem and to approximate the full-size eigenvectors solution for ISA [153]. Nyström and random feature based techniques, however, typically use random subsampling which may lead to choosing a subset of data points that do not represent the underlying structure of the data. Thus, *data selection* becomes of great importance to find a proper subset being representative of the full data set [94, 95].

Thus, we aim at selecting a subset, $\mathcal{S}$, from the full data set, $\mathcal{S}_{full}$, with much smaller number of data points and well representative of the structure of data. To properly subsample from the full data set, we utilize a scheme based on the

quadratic Renyi entropy [1] which is defined as:

$$E = -\log \int \mathbf{P}(z)^2 dz. \tag{4.6}$$

for a continuous random variable z with probability density function (PDF) $\mathbf{P}(.)$. To that end, a subset $\mathcal{S}$ including m data points is selected such that $E(\mathcal{S})$ is maximized. This entropy can directly be estimated from the samples using the kernel (Parzen) estimate of the PDF [48, 113]. Suppose the m data points are independent and identically distributed (i.i.d.) samples from a random variable; the Parzen estimate of the PDF using an arbitrary kernel function $\kappa(.,.)$ is given by [108]:

$$\hat{\mathbf{P}}(x) = \frac{1}{m} \sum_{i=1}^{m} \kappa(x, x_i). \tag{4.7}$$

The most widely used kernel in this context is a Gaussian one: $\kappa(x, y) = exp(-\|x - y\|^2/2\rho'^2)/\sqrt{2\pi}\rho'$. Thus, considering $\hat{\mathbf{P}}(x)$ as an approximation of $\mathbf{P}(x)$, it can be shown that:

$$E(\mathcal{S}) \approx -\log \int \hat{\mathbf{P}}(x)^2 dx = -\log(\frac{1}{m^2} \mathbf{1}_m^T \mathcal{K} \mathbf{1}_m) \tag{4.8}$$

$\mathbf{1}_m$ is a vector of m ones and $\mathcal{K}$ is the $m \times m$ Gaussian kernel matrix with the kernel size $\rho = \sqrt{2}\rho'$ also called Parzen window size. It is worth noting that the kernel matrix in (4.8) is shown with a different notation from equation (4.5) to avoid confusion between these two sets of kernels which play a very different role. Assuming ρ is chosen properly, the term $\mathbf{1}_m^T \mathcal{K} \mathbf{1}_m$ has larger values when data points are close to each other and form a region of high density which is indicative of a bad subsampling as selected data probably does not cover all regions. Therefore, having a better subset representing the underlying structure of all data equates to smaller values for the term $\mathbf{1}_m^T \mathcal{K} \mathbf{1}_m$, so the entropy should be maximized to ensure that the selected subsamples are spread over the entire data region and not only concentrated on a certain area of the data set.

$E(\mathcal{S})$ can be maximized iteratively in a greedy manner in order to select points that preserve the underlying structure of the data [144]. To accomplish this, the following algorithm is used:

1. Randomly select a subset $\mathcal{S}$ from the full data set $\mathcal{S}_{full}$.
2. Compute the quadratic Renyi entropy of $\mathcal{S}$ using (4.8).
3. Select a data point x^* from $\mathcal{S}$ and select a data point x^{**} from the remaining pool of data $\mathcal{S}_{full} \setminus \mathcal{S}$.

[1] Renyi entropy is chosen because it is easy to represent it with a Kernel matrix.

4. Replace x^* with x^{**} and compute the the quadratic entropy of the new subset.
5. If the entropy in step 3 increases compared to the entropy of $\mathcal{S}$, then x^* and x^{**} are swapped; otherwise they return to their first subsets.
6. Iterate from step 3 ...

The aforementioned algorithm can also be modified such that instead of swapping data points one by one, a mini-batch of data points can be replaced in each iteration.

To further illustrate the importance of data selection and the effect of the entropy-based method, we examine a toy artificial data set. This data set is represented in 3-dimensional space with a helix structure as shown in Figure 4.1-(a). Then, to form the $\mathcal{S}$ subset, 30 data points are selected. They are highlighted with rounded red points associating with edges after constructing 3-nearest neighbor graphs. Figure 4.1-(b) shows the results of the random selection; it is obvious that this subset does not represent the helix structure properly as the selected data points are not uniformly distributed. Using this subset as the initial one and applying the above algorithm, the new subset of data points achieved after 2000 iteration is shown in Figure 4.1-(c); this new subset is clearly representing the underlying structure of original data set better. However, entropy-based data selection can yield bad subsampling if ρ is not chosen properly. Figure 4.1-(d) shows such a case where ρ is too large to preserve the local structure of data.

Good kernel density estimation requires the selection of the proper Parzen window size. The same holds for the intended entropy maximization. A variety of methods have been proposed to select ρ such that $\int[\mathbf{P}(x) - \hat{\mathbf{P}}(x)] \approx 0$ [29]. The most effective way to set this hyperparameter may be to cross-validate it; this, however, does add further computation since ISA hyperparameters need to be cross-validated too. To alleviate the computations, Silverman's rule [138] can be used as a rule of thumb:

$$\rho_{sil} = \delta[\frac{4}{(2D+1)n}]^{[1/(D+4)]} \tag{4.9}$$

Where D is the dimension of data, δ is the sum of diagonal elements in the covariance matrix of data in $\mathcal{S}_{full}$, and n is the number of data points in $\mathcal{S}_{full}$. We will examine both Silverman's rule and cross-validation.

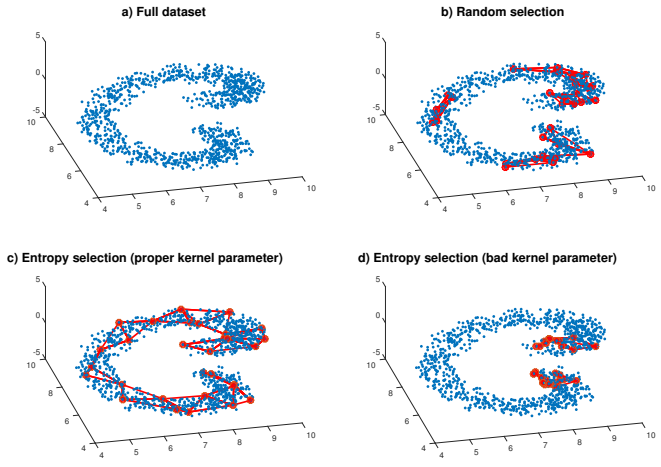


Figure 4.1: Maximizing the quadratic Renyi entropy to select more representative points for a toy data set with a helix structure.

4.5 Experiment 1: Word Recognition

4.5.1 Setup

This section describes word recognition experiments on 5 languages taken from the GlobalPhone dataset: German (GE), Spanish (SP), Portuguese (PO), Russian (RU) and French (FR). In our setting, we simulate low resource conditions by constructing randomly selected training sets of data containing 1 hour (8 speakers) using 7-8 minutes of recorded speech for each of the selected speakers in each language. HMM/GMM systems were trained in the same way as explained in Section 4.6.1. For the DNN training, however, ReLU nonlinearity is employed. We also conducted several experiments to compare the performance of ISA with random data selection and the entropy-based one using different settings.

4.5.2 Results

First, we focused on the German data and explored the effect of subset size and Parzen window size on the results. Without loss of generality, in this set of experiments we used monophone systems to mitigate the computational

load. We considered three settings including 1k, 2k and 5k data points to learn ISA features. For each setting, both random and entropy-based data selection were investigated. Furthermore, we examined two scenarios for entropy-based data selection where Parzen window size was either tuned by a cross-validation, ρ_{opt}, or obtained by the Silverman's rule, ρ_{sil}. In our experiments, we always used 2000 iterations for entropy-based data selection algorithm presented in Section 4.4 and the mini-batch size is $0.1m$ (m is the subset size). Figure 4.2 shows the comparison for the results based on 5 runs. The baseline result shown in Figure 4.2 corresponds to the MFCC features.

There are a number of trends apparent in Figure 4.2. The data selection method using cross-validation to find the Parzen window size (ρ_{opt}) clearly and consistently outperforms the one with the Silverman's rule (ρ_{sil}) and random selection. Moreover, ISA features using proper data selection in the construction of the Laplacian matrix outperform the traditional MFCC features. Besides, we expect to have better results by increasing the number of data points; this is the case for the entropy-based data selection with ρ_{opt}; nonetheless, the random data selection with the subset size of 2k does not result in a better performance than 1k while the one with 5k clearly improves the results; this can be simply attributed to the randomness in the data selection. Another interesting observation can be made by comparing the performances of ISA using the entropy-based data selection with ρ_{opt} for the subset sizes of 2k and 5k. The variation of the WERs in both cases is very small and the results are close to each other; this implies that as long as Parzen window size is chosen properly, 2k and 5k data points can provide almost equally good subsets. A reasonable question which might arise is that how big we should choose the subset to fill the gap between the random selection and the entropy-based one. This depends on the size and structure of the original data. In this setting, we also conducted some experiments with subset size of 10k and surprisingly we observed that WERs were higher than those obtained with 5k subset size. This is the same behavior we observe in Figure 4.2 for 1k and 2k subset size. The entropy-based data selection with ρ_{opt}, however, was still in the same range as for the subset size of 5k.

Next, we extended our experiments by using a standard HMM/GMM system. Table 4.1 compares the WERs for baseline MFCC with ISA being trained either with the random data selection or the entropy-based one. The entropy-based data selection in this table is the one which uses ρ_{opt} and $m = 5k$. For each language, the total number of Gaussians and tied-states were tuned using the development sets and WERs for both development (Dev) and evaluation (Eval) sets are presented. It can be observed that in all cases, ISA trained on entropy-based selected subset performs the best. Moreover, for some languages like PO and RU ISA with random selection degrades the performance compared

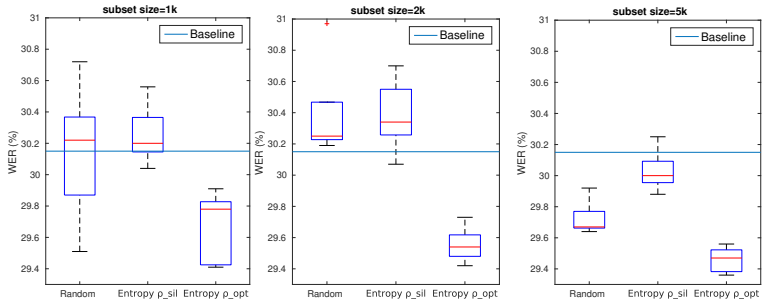


Figure 4.2: Boxplot of WER on development set (5 runs) using ISA trained by random and the entropy-based data selection for different subset size. Results are given for the 1hr of German training data. The baseline results are obtained with MFCC features.

Table 4.1: Comparing WERs using MFCC and ISA features using both random and entropy-based data selection in HMM/GMM systems for five languages with 1hr of training data.

		GE(1hr)		FR(1hr)		SP(1hr)		PO(1hr)		RU(1hr)	
		Dev	Eval	Dev	Eval	Dev	Eval	Dev	Eval	Dev	Eval
MFCC		22.84	35.38	37.28	34.91	35.33	20.02	46.39	47.52	48.45	47.01
ISA	Random	22.72	34.92	37.71	34.86	34.69	19.08	46.57	47.65	49.65	47.31
	Entropy	**22.31**	**34.12**	**36.80**	**34.03**	**34.09**	**18.95**	**47.23**	**47.21**	**48.13**	**46.24**

to the MFCC while for GE and SP the performance is improved. However, we should note that we conducted the experiments only one time and there is no guarantee that the selected random subset would lead to the same results if we repeated the experiments again, whereas we believe the entropy-based data selection shows a more robust behavior.

We also experimented with DNNs, but found that DNNs generally underperform HMM/GMM systems in the low resource scenarios that we are investigating - contrary to their superior performance when more training data is available. Thus, we considered two other settings for German dataset with 14.85hr and 5hr of data. Table 4.2 shows the WERs for these settings using ISA features compared to conventional features. Note that unlike HMM/GMM systems, FBANK features are better inputs for HMM/DNNs; thus, our baseline systems for HMM/GMM and HMM/DNN training used MFCC and FBANK features respectively. We can observe that only in HMM/GMM training ISA outperforms baseline system while for HMM/DNNs either no or marginal improvement is obtained. Yet, it can be seen that the entropy-based data selection still shows the robust behavior. This implies that when reasonable amounts of data exist,

Table 4.2: WERs for both GMM and DNN systems using 5hr and 14.85hr of German data. ISA with/without the entropy-based data selection is compared with the traditional features.

systems		HMM/GMM			HMM/DNN		
Features		MFCC	ISA		FBANK	ISA	
			Random	Entropy		Random	Entropy
5hr	Dev	15.70	15.75	15.47	13.40	13.88	13.41
	Eval	24.41	24.50	24.17	22.93	23.07	22.97
14.58hr	Dev	13.95	14.43	13.87	11.85	11.82	11.78
	Eval	21.36	21.40	21.30	19.49	19.48	19.29

it is beneficial to exploit all the nonlinearities in a discriminative manner with a DNN rather than applying the unsupervised manifold based transformation before the input layer.

4.6 Experiment 2: Phone Recognition

4.6.1 Setup

In this set of experiments, ISA is compared with conventional features for the Afrikaans data as introduced in Section 2.5. To extract ISA features, the FBANKs were used and 10k samples were randomly selected from the training data to construct the weighted similarity graph and consequently make the normalized graph Laplacian; noting that in this experiment we only used random data selection and in the next section the entropy-based data selection is employed in the experiments. Our ISA definition involves four hyperparameters which were jointly optimized on the validation set introduced in Table 2.2. Suitable parameters were tuned using the Dev. set as follows: $\epsilon = 5$, $\sigma = 90$, $\xi = 1$, $\tau = 0.5$. It was observed in the experiments that tuning these parameters using different amounts of data or based on different systems, e.g. monophones, triphones or SGMMs, leads to almost the same values.

For acoustic modeling, we first trained conventional 3-state left-to-right HMM triphone models. Subsequently, the alignments of the triphone system were used to train SGMMs. In these experiments, MFCC, PLP and ISA as well as FBANK features were used. The raw features together with their first and second derivatives were spliced in time taking a context size of 7 frames (i.e.,± 3), followed by decorrelation and dimensionality reduction to 40 using LDA and

Table 4.3: PERs(%) with different total number of Gaussians for HMM/GMM systems trained on 1hr of Afrikaans.

# of Gaussians	Features			PER(%)
	MFCC	FBANK	ISA	
Total/ Average per tied-state	**4k/9.2**	2.2k/4.8	1.8k/3.9	23.09
	-	**4k/8.8**	3.2k/7.0	22.22
	-	-	**4k/6.6**	21.79

further decorrelation using MLLT [44]. Also, we trained HMM/DNN systems using a generalized maxout network with 2-norm nonlinearity.

4.6.2 Results

A key motivator is the expectation that fewer Gaussians will be needed for acoustic modeling of the intrinsic features rather than the extrinsic ones. To investigate this issue, we first consider 1hr of Afrikaans data, and then tune the number of Gaussians using the validation set. The PERs for MFCCs, FBANKs and ISAs are 23.09%, 22.22% and 21.79% respectively and are shown in the last column of Table 4.3; also, the tuned values for the total number of Gaussians and their average per tied-state are made bold. As shown, ISA outperforms MFCC and FBANK features.

Table 4.3 reveals more interesting trends; first, we can see that fewer Gaussians (on average) are required to model triphones with ISA features than MFCCs and FBANKs. Moreover, for each feature type, the number of Gaussians needed to have the same PERs corresponding to the other features were found. For example, to have a PER equal to 23.09% we need only ∼1.8k Gaussians using ISA which is less that half of those required for MFCC. Table 4.4 summarizes the PERs for the all training sets introduced in Table 2.2 for MFCC, PLP, FBANK and ISA features. In each scenario the number of Gaussian components is tuned first. It confirms our hypotheses that ISA features outperform the other feature representations since fewer components are needed to effectively train phone models in both the HMM/GMM and SGMM systems.

In addition, it is of interest to contrast ISA with other conventional features as input for DNNs. We compared ISA with FBANK and MFCC features. PLP features were not included in this experiment as we observed that their performance was very similar to MFCCs, and they were never better than FBANK features. The number of hidden layers were set based on the amount of training data. The number of units in each hidden layer and the group size were 800 and 5 respectively. Table 4.5 shows the PERs for various features

Table 4.4: Comparing PERs(%) using different amounts of Afrikaans training data for FBANK, MFCC, PLP and ISA in monolingual GMM based systems.

Systems	Features	Train1	Train2	Train3
HMM/GMM	FBANK	22.22	16.13	13.91
	MFCC	23.09	16.87	14.81
	PLP	23.41	16.77	14.80
	ISA	**21.79**	**15.75**	**13.55**
SGMM	FBANK	21.65	13.07	10.53
	MFCC	22.38	13.83	11.02
	PLP	21.97	13.42	10.83
	ISA	**21.29**	**12.84**	**10.42**

Table 4.5: Comparing PERs(%) using different amounts of Afrikaans training data for FBANK, MFCC, PLP and ISA in monolingual HMM/DNN systems.

Set		Train1	Train2	Train3
# of hidden layers		2	3	4
Feature (#dim)	FBANK(24)	23.47	**13.95**	**11.27**
	MFCC(13)	23.30	15.10	12.67
	MFCC(24)	24.06	14.96	12.56
	ISA(13)	**22.39**	14.37	12.14
	ISA(24)	23.62	14.12	11.91

with different dimensions using HMM/DNN systems. The number of DNN targets (i.e. context-dependent triphone states) were 505, 1380 and 2281 for Train1, Train2 and Train3 respectively. For Train1 set, we repeated the DNN training three times and reported the average of the results. As is shown, DNN performances are worse than SGMM systems, noting that SGMMs reduce the number of parameters by choosing the Gaussians from a subspace spanned by a background model while we need to train many parameters for DNN systems. Moreover, except for the Train1, FBANK outperforms ISA in the monolingual DNNs systems. This suggests that when a reasonable amount of data exists, it is beneficial to exploit all the nonlinearities in a discriminative manner with a DNN rather than applying the unsupervised manifold based transformation before the input layer.

4.7 Conclusions

In this chapter, we proposed to use ISA features which are representative of speech articulatory configuration parameters in low resource conditions. We successfully showed that this allows acoustic modeling with a smaller number of parameters in GMM based acoustic modeling. Moreover, we investigated the impact of data selection on the performance of ISA; we proposed to use a data selection method by maximizing nonparametric estimation of the quadratic Reniy entropy to achieve a subset being well representative of the original full dataset. We conducted several experiments and we observed that ISA is a good alternative to the conventional features in low resource settings specially for GMM based acoustic modeling. We also observed that when 5hr or more data is available, ISA is not advantageous for DNN-based recognition.

Chapter 5

Speech Manifold for Crosslingual and Multilingual ASR

This chapter is adapted from the following article(s):

- Reza Sahraeian and Dirk Van Compernolle. Crosslingual and multilingual speech recognition based on the speech manifold. Accepted to be published at IEEE/ACM Transactions on Audio, Speech, and Language Processing.

5.1 Introduction

This chapter describes our investigation of using ISA for crosslingual and multilingual acoustic modeling. In addition to what was explained in Section 2.4, another approach to achieve cross-language portability is the use of articulatory features instead of acoustic features. Articulatory and similarly phonological features are appealing for speech and language technology as they are considered spoken language universals [20, 70]. However, the acoustic-to-articulatory inversion is a non-trivial problem. The mapping can be learned explicitly using dedicated articulatory databases [107], or it is learned implicitly by fitting an articulatory based speech synthesis model [64, 106, 118]. Alternatively, the mapping is approximated as a detection of phonological feature bundles [78, 140, 141]. All of these systems have considerable weaknesses such as very expensive databases, mathematical issues in dealing with the model inversion and low reliability of the phonological feature transcriptions of the training data. So, while conceptually appealing and some improved language independence can be demonstrated, acoustic-to-articulatory mapping has not yet become a mainstream component of speech recognition systems.

In this chapter, we also aim to benefit from *articulatory-like* features in crosslingual and multilingual scenarios, but avoiding the pitfalls associated with an explicit model for acoustic-to-articulatory inversion. Our work is based on Intrinsic Spectral Analysis (ISA) as described in Section 4.3.2; ISA has shown a great promise as a feature transformation, and given the correlation of several of its coordinates to individual articulatory features [74], we consider the ISA representation a worthwhile candidate for multilingual or crosslingual ASR settings. While training ISA coordinates on one language and using them for another language can simply represent a crosslingual scenario, the multilingual training of ISA is more challenging with regards to data selection and exploiting the local structure of data. We propose how to efficiently select data from each language and construct graphs that are representative of the structure of the data.

Then, since DNNs have emerged as the prevailing paradigm in speech recognition today and have shown a great success in providing language independent features [158, 162], we investigate the usefulness of the universal feature space that ISA provides in combination with crosslingual and multilingual DNNs. We first examine the feature representation of hidden layers in crosslingual scenarios, reminiscent of the monolingual settings studied in [97, 101, 174]. Intuitively, we may believe that the low level and local characteristics which are taken care of by the lower layers in a DNN are likely to be less language dependent; however, in this work, we go beyond intuition and demonstrate that hidden layers show almost the same selectivity to broad phonetic classes of the speech sounds no

matter what language they are from. This finding suggests that hidden layers to some extent form a language independent acoustic-to-phonetic mapping. Since the objective function in DNNs does not relate to the manifold structure of speech, we argue and demonstrate that using the manifold learning scheme together with DNNs leads to improved language independence.

5.2 Multilingual ISA

The main goal of this chapter is to employ ISA in multilingual and crosslingual speech recognition. Crosslingual ISA can be simply defined as training f (in (4.3)) on one language and applying it to another language. In the multilingual training of ISA, however, we utilize data from multiple languages to train a universal mapping f_{univ}. This implies that we need to construct the adjacency graph with multilingual data in the acoustic space. This immediately raises a number of questions in terms of data selection and hyperparameter tuning. This section describes these issues together with our proposed approaches to train efficient and reliable multilingual intrinsic coordinates.

The first question that arises is how to select data from the multilingual pool of data to form the graph. A reasonable approach seems to select random subsets per language of roughly the same size. In the case of many languages an optimization conflict arises between enough samples per language to have a representative set and not having too many samples in the graph to keep the computation of the Laplacian manageable. To handle this issue, we can benefit from the entropy-based data selection method explained in the Section 4.4.

Next, the adjacency graph should represent the structure of the data; however, data from different languages may exhibit various structures that originate from natural variabilities across languages. This suggests to construct the graph for each language individually using the selected data from that language. In other words, for any language s, the similarity matrix $\mathbf{W}_s$ is generated by using hyperparameters ϵ_s and τ_s. Thus, these graph hyperparameters are tuned for each language to ensure that the graph represents the structure of data for that language properly. Then, we form the multilingual graph from the set of monolingual graphs so that the intrinsic coordinates obtained from (4.4) can be used universally for any language. Constructing the multilingual graph requires both inter-language and intra-language similarity information; For sake of simplicity and in order to keep the number of hyperparameters limited, we decided to ignore the inter-language term so that the multilingual similarity

matrix takes a block diagonal form:

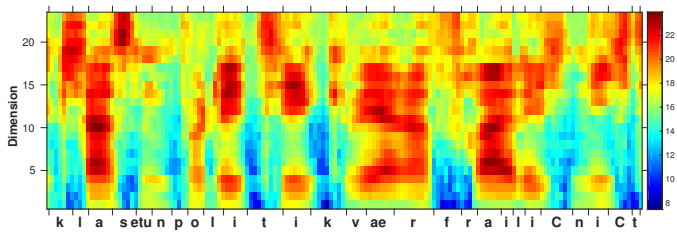
$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & 0 & \dots & 0 \\ 0 & \mathbf{W}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{W}_S \end{bmatrix} \tag{5.1}$$

Based on equation (4.1), this means if x_i and x_j are close to each other but not from the same language, we don't insist on keeping them close after mapping; at the same time there is nothing in the ISA algorithm that forces them apart given zero-affinity in the affinity matrix. Thus, although this assumption might ignore some information, it saves us a lot of computations in terms of both hyperparameter tuning and eigendecomposition as the Laplacian matrix becomes very sparse. Furthermore, the Parzen window size for entropy-based data selection can be tuned together with graph hyperparameters for each language individually and thus we are only left with σ and ξ which need to be tuned multilingually.

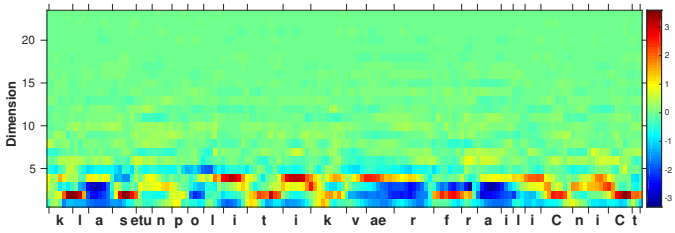
There is one further concern relating to the block diagonal structure of the multilingual graph Laplacian matrix. With such a Laplacian matrix, the common LE framework would give S eigenvalues equal to 0. The corresponding eigenvectors are indicators of languages and separate the training data by language. The rest of the eigenvectors are the eigenvectors of individual subgraphs which contain 0's in dimensions relating to the other blocks. So, when using a block diagonal affinity matrix, there is no sharing of information across languages in the intrinsic dimensions in the LE framework. This is the opposite of what we want, i.e. an intrinsic space that is applicable to all languages. However, the situation is very different when using ISA. Thanks to the manifold regularization the separation between the subgraphs is smoothed out in the kernel space as there is substantial overlap between the speech features of the different languages in acoustic space. This implies that although the subgraphs are disjoint, intrinsic dimensions are obtained that can be shared across languages. We observed that all dimensions of the ISA representation contribute significantly to all languages. This may be illustrated by computing the total variance along an intrinsic dimension for any language. E.g. for the first intrinsic dimension the largest standard deviation is only 41.7% bigger than the smallest one.

5.3 Dimensionality of ISA

Manifold learning techniques, including ISA, seek for a low dimensional representation of data and is mostly used as a dimensionality reduction method.



(a) FBANK



(b) ISA

Figure 5.1: Comparing ISA representation with FBANK for a segment of speech data in German: "klassenpolitik wäre freilich nicht".

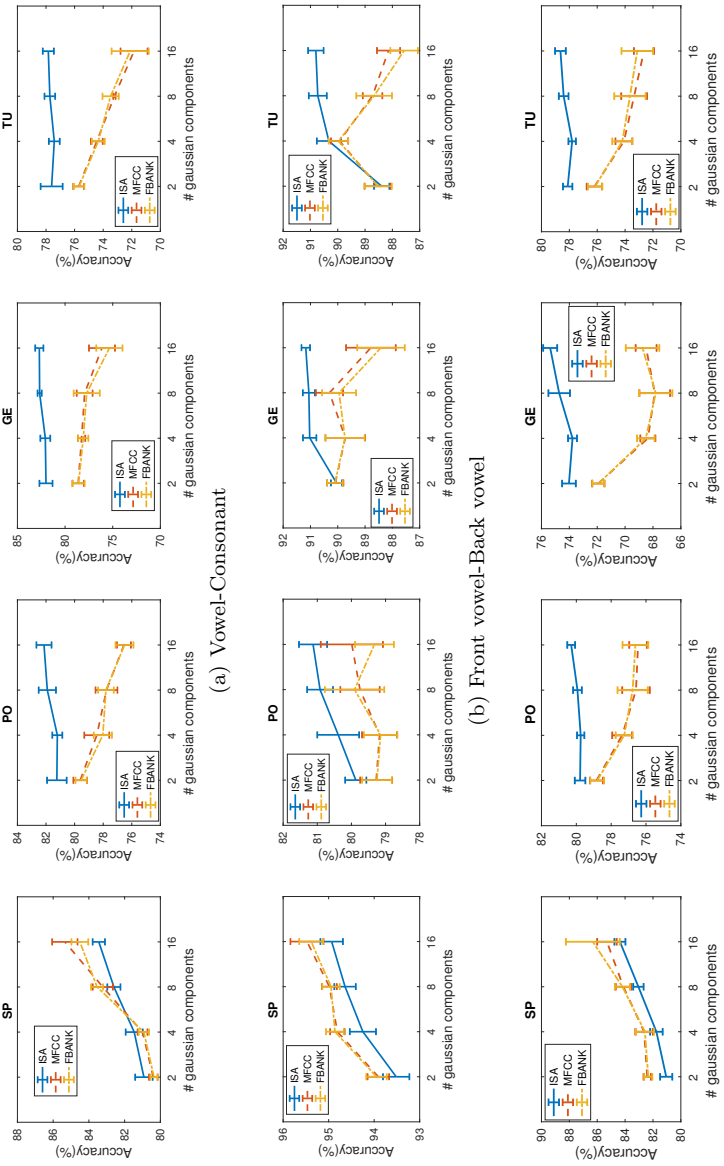
In our case we expect ISA to learn articulatory-like features which are expected to be low dimensional. To inspect these characteristics and to have a feeling for how ISA features look like, ISA features as well as the corresponding FBANK ones are presented in Figure 5.1 for a segment of speech in German; for this example the intrinsic coordinates were trained multilingually. It is clear in Figure 5.1 that the discriminative information is mostly contained in the lower eigenvalued components. This means that we probably could ignore many of the higher eigenvalued components in our experiments with limited loss in performance; however, in this chapter for the crosslingual and multilingual ASR, we used the same dimensionality for ISA as FBANK only to be consistent in the experiments as changing the dimensionality changes the model size. Besides, what we are mainly interested in the connection of ISA to the articulatory configuration space and how it might lead to language independence.

5.4 Cross-language Portability of Intrinsic Coordinates

The idea of using ISA for crosslingual and multilingual scenarios arises because we expect a correlation between intrinsic coordinates and articulatory features which should show less language dependence. In this section frame-based binary classification tasks are investigated in which we compare the performance of ISA with conventional MFCC and FBANK features in terms of language independence.

The multilingual data used for these experiments includes Spanish (SP), Portuguese (PO), German (GE) and Turkish (TU). For each language 10k utterances were randomly selected from the training portion of the datasets. First, we considered SP as the training language and the others for test to investigate cross-language portability of GMM classifiers for broad phonetic classes that were trained using SP data; we also present the monolingual case where test data is taken from SP. GMMs are trained with full covariance matrix and all features have the same dimensionality of 23. In each scenario 5k data points were randomly selected for each class and the experiments were repeated 5 times to ascertain that results are statistically significant. Apart from MFCC and FBANK features, ISA features for the test languages were derived from the intrinsic coordinates trained on SP data. In other words, the function f was fully estimated from SP data in a monolingual manner and the ISA hyperparameters $(\epsilon, \tau, \xi, \sigma)$ were tuned using validation data from SP.

Figure 5.2 shows the classification results for three class pairs: vowels vs consonants, front vowels vs back vowels and fricatives vs non-fricative consonants. Included in each plot are the results for different number of Gaussian components. Figure 5.2 reveals the following trends: first, in the monolingual case, i.e. test is also SP, ISA does not outperform FBANK or MFCC; however, in the crosslingual settings, ISA clearly outperforms the others. Moreover, as the number of Gaussian components increases, FBANK and MFCC performance mostly degrades. This can be attributed to overfitting to the training language. However, interestingly, ISA performance is not affected in the same way. This suggests that the feature distribution of these broad phonetic classes in the ISA feature space is more portable across languages. These results suggest that ISA provides an acoustic representation with less language dependency compared to conventional FBANK and MFCC features. This figure also suggests that in the monolingual setting ISA does not necessarily outperforms the other features; this seems to be in contrary to what we got in the previous chapter. However, we should note that the setting is different and no entropy based data selection is applied here.



(a) Vowel-Consonant

(b) Front vowel-Back vowel

(c) Consonant fricative-Consonant non-fricative

Figure 5.2: Comparing the performance of ISA with MFCC and FBANK features in three binary frame classification tasks. GMM models are trained on SP and test languages are specified on top of each plot.

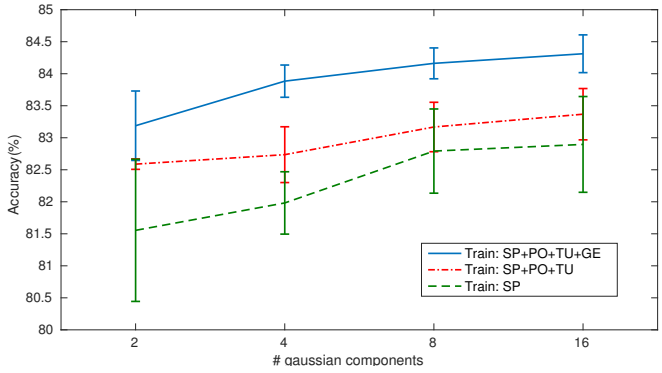


Figure 5.3: Comparing the performance of ISA trained on different languages for vowel vs consonant classification task on GE.

Next, we examined two cases where ISA was trained multilingually. For these experiments we used GE as the test language; in the first scenario, data from SP, PO and TU were merged to construct the graph language independently and learn the intrinsic coordinates. In the second one, GE data was included as well. For the vowel-consonant classification task, Figure 5.3 compares the performance of intrinsic coordinates being learned from different language combinations. The best performance is obtained using multilingual training data including GE, which is the target language, to derive intrinsic coordinates. It is also notable that ISA being trained on SP, PO and TU outperforms the one trained with only SP; although GE was not used to construct the Laplacian graph and none of the training languages was closely related to GE, it seems that ISA benefits from the generality gained by merging languages.

The results of the experiments in Figure 5.2 and Figure 5.3 suggest that while ISA provides a less language dependent feature space than FBANK and MFCC, a set of universal intrinsic coordinates trained from multiple languages is beneficial for both seen and unseen target languages. Moreover, in the view of the connection between individual intrinsic coordinates and distinctive features explored in [74], it is of interest to look at this characteristic in a multilingual case as well. To that end, we used the multilingual ISA trained on SP, PO, GE and TU from the above experiment. Under the assumption that ISA provides a mapping to articulatory-like features, we expect to observe that single ISA coordinates are particularly discriminative for a single acoustic-phonetic feature. In Table 5.1, we compare the classification results obtained from the most discriminative coordinate with the full-dimensional ISA(1:23). For example, we observed that the first coordinate is the most discriminative one for vowel vs consonant

Table 5.1: Classification accuracy using 23 intrinsic coordinates and the most discriminative one. ISA is trained multilingually using GE, SP, TU and PO.

Test Language		SP	PO	GE	TU
Vowels vs consonants	ISA(1:23)	79.40	81.86	83.32	79.30
	ISA(1)	71.99	72.81	77.48	72.76
Front vowels vs back vowels	ISA(1:23)	92.53	88.88	92.70	90.23
	ISA(4)	77.94	77.19	81.12	77.39
Consonant fricatives vs non-fricatives	ISA(1:23)	80.42	80.90	76.98	79.15
	ISA(2)	76.38	78.49	76.22	77.48

classification task no matter which language is used for test. It is worth noting that the second most discriminative coordinate in this case yields at least 10% lower (absolute) accuracy. We also observed that the 2nd component provides a good separation between fricatives and non-fricatives, and it can be seen that the classification accuracy of ISA(2) is very close to ISA(1:23). Note that in these experiments we used the same number of Gaussian components for the GMM classifiers in both 1-dimensional and 23-dimensional settings which is not optimal; however, we observed that tuning the number of Gaussian components results in less than 2% absolute improvements in all scenarios. It is also worth mentioning that as pointed out in [74], in monolingual settings, some of the natural classes can be discriminated reasonably well via extrinsic components too. In the multilingual experiments, while we observed the same behaviour for some phonetic classes, usually there is a group of dominant discriminative coordinates (and not only one). Also the results are never better than ISA; for example in the vowel-consonant classification using MFCC(1), which is the most discriminative coordinates in this case, gives an accuracy of 70.02%, 69.72%, 74.33% and 69.45% for SP, PO, GE and TU respectively which are all significantly less than what we obtained with ISA(1).

5.5 DNN Hidden Layers as Universal Feature Extractors

So far, sets of binary classification experiments were conducted to demonstrate that ISA can yield a feature space in which phonemes from different languages share more similar distributions than spectral-based features. In the rest of the chapter we explore the usefulness of ISA coordinates as inputs to DNNs for crosslingual and multilingual speech recognition. As we observed that FBANK features consistently outperform MFCCs as input features for DNNs, we did not

include any results with MFCCs for any of the DNN experiments and restricted ourselves to a comparison of FBANK and ISA. Multilingual DNNs rely on the assumption that the hidden layers are universal feature transformations that can be shared across languages. Therefore, the hidden layers are trained simultaneously for different languages such that they can benefit from a larger and more diverse pool of training data. The multilingual hidden layers, with an optional bottleneck layer, can be employed to extract features that will serve as input to another GMM or DNN based system, e.g. [149, 158], or they can be reused in a language specific DNN where only the softmax layer is trained on the target language [65]. No matter what configuration is being used, we want to find out if using ISA brings additional language independence.

5.5.1 Feature Learning in Crosslingual DNN

Inspired by the recent work which explores node selectivity to specific phonetic features in a DNN [101], we investigate how hidden layers characterize the input features and why they can be transferred successfully between languages. Towards this end, we look at the representational properties of the DNN output features with respect to the phonemic categories. For the sake of visualisation, we consider a bottleneck feature extractor with dimensionality 40. The configuration of the bottleneck feature extractor is shown in Figure 2.6.

We used the training portion of Spanish (SP) to train a 6 hidden layer network. Each layer consists of 300 hidden units except the fifth layer which is the bottleneck layer; the number of target context-dependent states was set to 3100[1]. DNN inputs were ISA features derived from the intrinsic coordinates trained by 10k data points randomly selected from the Spanish training data; the ISA hyperparameters were also tuned monolingually on the Spanish validation set: $\epsilon = 5$, $\sigma = 200$, $\xi = 1$, $\tau = 5$. We are interested in the response properties of individual nodes in the bottleneck layer. Since the activation function is ReLU, the bottleneck feature values can be very large, therefore we first normalized (z-scored) the bottleneck features. Then, at each node we calculated the mean response of the node to each phonetic category; we used 7 categories based on manner and place of articulation (front vowel, back vowel, open vowel, plosive, labial, nasal, fricative) . Node responses of the bottleneck layer are shown in Figure 5.4 for these categories in SP, PO and GE.

The top plot in Figure 5.4 shows the monolingual behavior for SP. It can be observed that nodes exhibit different degrees of selectivity to the phonetic features. For example, node 1 and 30 show significant responses to back vowels only, while the 6th and 33rd nodes respond clearly to fricatives. These

[1]These are tuned on the Spanish development set.

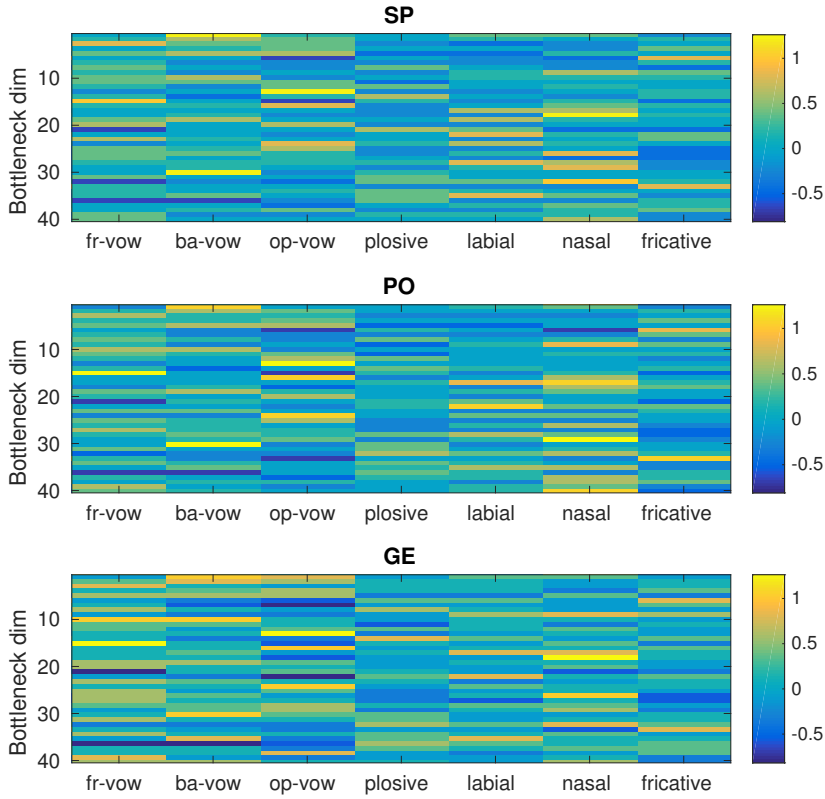


Figure 5.4: Node selectivity of the bottleneck layer for different phonetic features in mono- and crosslingual settings. ISA and the bottleneck feature extractor are trained on SP and test languages are shown on top of each plot.

observations are in line with [101] and reaffirm the node selectivity to the phonetic features. However, what we are specifically interested in is the crosslingual behavior. The other two plots in Figure 5.4 reveal the selectivity pattern of the same bottleneck nodes when the DNN input features are from the unseen languages PO and GE. Interestingly, there are clear similarities in the way the nodes respond to the phonetic features for different languages, while at the same time these similarities are more apparent between SP and PO which are closely related languages. For example, nodes 13, 16 and 24 always respond

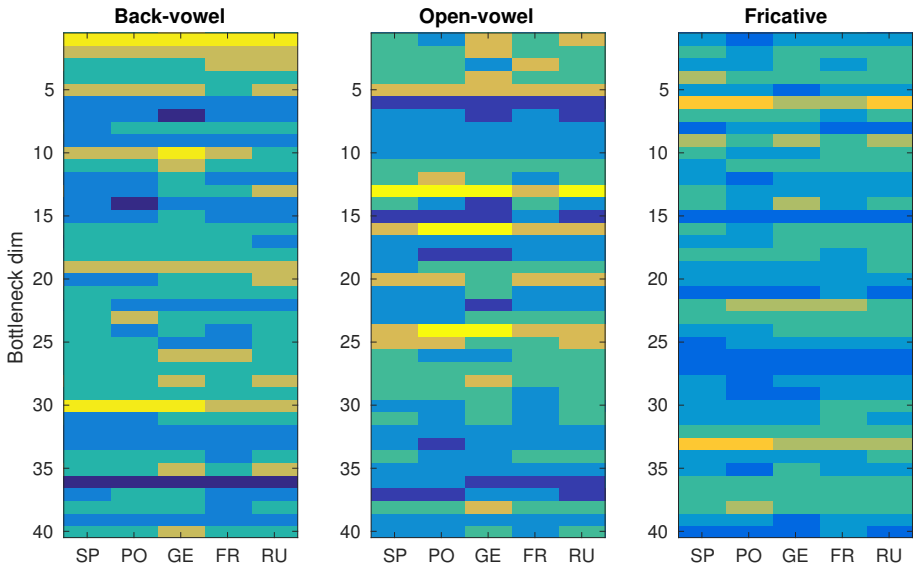


Figure 5.5: Node selectivity of the bottleneck layer for three phonetic feature categories for different languages for ISA and DNN trained on SP.

to open vowels, and nodes 6 and 33 respond to fricatives in all cases. Therefore, the fact that nodes respond to distinctive features which are generic units of speech representation may suggest success in cross-language knowledge transfer. For the sake of comparison, the selectivity for three phonetic feature categories is displayed separately in Figure 5.5 including French (FR) and Russian (RU) as well. The same observation can be made that individual or small groups of nodes are responsive to specific phonetic features, and this property appears to be relatively stable across languages. Finally, it is worth mentioning that we observed the same behavior from a multilingual DNN.

5.5.2 ISA vs FBANK

The results from Figure 5.4 and Fig. 5.5 suggest that a DNN bottleneck feature extractor trained on ISA implements an acoustic-to-phonetic transformation which shows a high degree of language independence. For comparison we also investigated if DNNs trained on conventional FBANK features exhibit the same property across languages. While the same property can be observed to some

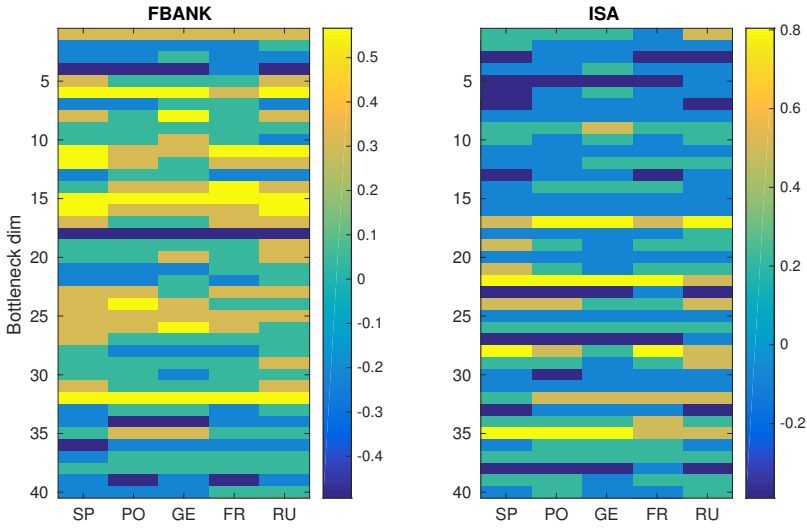


Figure 5.6: Node selectivity to labial consonants of different languages for DNNs trained on FBANK and ISA (for SP).

extent, there are some differences. For example, in Figure 5.6 the node responses to labial consonants are presented for various languages for DNNs trained on FBANK and ISA. It can be observed that even with FBANK features there are some specific nodes activated across languages; however, with ISA features the node selectivity is more apparent. Moreover, the confusability of nodes across phonetic features is less with ISA features. For example, we observed when the DNN is trained on FBANKs, some of the active nodes for labials also respond to other classes whereas for the DNN trained on ISA, the active nodes respond to labials more distinctively.

The fact that ISA may further improve the language independence of a DNN bottleneck can be attributed to two properties. Firstly, DNNs have a supervised training process where the objective is to discriminate between output targets; in a multilingual scenario, for example with multi-task learning [65], shared hidden layer parameters are forced to learn the best possible discrimination for all tasks. Now, as ISA training is subject to a quite different objective, i.e. to preserve the local structure of data in an unsupervised manner, DNN and ISA can be complementary methods. Secondly, a DNN trained on FBANK features does not provide a perfectly language independent acoustic-to-phonetic transformation. While similar selectivity patterns are to some extent observed

for various languages, there are still some confusions which naturally originate from between language variabilities. A possible way to remedy this problem is to train the DNN in a less language dependent feature space, and according to our achievement in Section 5.4 that ISA exhibits less language dependence compared to FBANK, DNNs trained on ISA might bring more language independence.

5.6 Experiments

The primary aim of this section is to compare ISA with commonly used FBANK features in the prevailing DNN based language independent approaches. The recognition task is a standard dictation task for a subset of languages of the GlobalPhone dataset.

5.6.1 Monolingual low resource baseline

First, for each of the low resource settings, we present in Table 5.2 monolingual baseline results obtained from different systems. The input features were MFCCs for the SGMM and HMM/GMM models and FBANKs for the DNNs. All features were mean and variance normalized (CMVN) and then being spliced in time taking a context size of 7 frames (i.e.,$\pm$3), followed by decorrelation and dimensionality reduction to 40 using LDA and further decorrelation using MLLT. We observed that for this low resource scenario SGMM outperforms GMM and DNN based acoustic models. The UBM used to initialize the SGMM included 200 components, and the number of leaves and substates were 2000 and 3000 respectively; these values were tuned empirically.

Table 5.2: Monolingual results (WER%) for the low resource settings (GE(1hr), PO(1hr), RU(1hr) and MAN(1hr)) using different systems.

Low resource target language		HMM/GMM	DNN	SGMM
GE(1hr)	Dev	22.84	21.41	19.03
	Eval	35.38	34.90	32.97
PO(1hr)	Dev	46.39	48.65	40.15
	Eval	47.52	51.85	40.76
RU(1hr)	Dev	48.45	51.70	44.54
	Eval	47.01	47.30	43.12
MAN(1hr)	Dev	51.76	52.55	45.06
	Eval	61.78	62.28	53.40

Table 5.3: Crosslingual WER(%) results using the bottleneck feature extractor trained with ISA and FBANK; SP and FR are donor languages and four low resource languages are tested.

Target Languages		Donor language: SP		Donor language: FR	
		FBANK	ISA	FBANK	ISA
GE(1hr)	Dev	18.97	**18.29**	18.81	**18.65**
	Eval	33.16	**31.61**	31.56	**30.94**
PO(1hr)	Dev	37.80	**35.69**	36.68	**35.87**
	Eval	40.82	**39.05**	39.99	**38.31**
RU(1hr)	Dev	43.66	**43.21**	44.57	**43.76**
	Eval	41.08	**40.53**	42.01	**40.84**
MAN(1hr)	Dev	41.97	**41.09**	41.03	**40.71**
	Eval	55.14	**52.60**	55.50	**54.17**

5.6.2 Crosslingual experiments

Next, we carried out two sets of crosslingual experiments. In the first one, Spanish plays the role of high resource donor language and in the second one French takes on that role. In each case, four unseen low resource target languages were examined. For these experiments a tandem approach was used in which a DNN trained on the high resource donor language provides bottleneck features as an input to an SGMM. The bottleneck features were then spliced in time and transformed with MLLT and LDA in the same way we did in the monolingual experiments. The DNN generating the bottleneck features is identical to the one used in Section 5.5.1. We considered two different feature representations: FBANK and ISA. ISA features were trained on the same high resource language as the one used for DNN training using 10k samples randomly selected from the training data.

The results in Table 5.3 reveal that in all cases ISA outperforms the FBANK features. This is a compelling achievement because in this set of experiments the intrinsic coordinates and the bottleneck feature extractor were trained on a single high resource donor language and no information of the target languages was available during training. Moreover, it is worth noting that we can observe that in some cases the tandem system with FBANK features does not outperform the SGMM monolingual results in Table 5.2. This can be ascribed to dissimilarity between donor and target languages. With ISA features, however, crosslingual ASR results are almost uniformly better.

5.6.3 Multilingual experiments

We used the full training data from all nine languages in Table 2.3. Given much more data, we are able to train a much larger multilingual network with 15 hidden layers and 1500 nodes per layer using multi-task learning so that each language has it's own output layer. In other words, the hidden and input layers were shared while each language had a dedicated softmax layer and the number of target context-dependent states was set to 3100 for each language (Figure 2.5). The recognition results are presented in Table 5.4 for six languages using different systems. For each target language, we utilized the multilingual hidden layers and the corresponding softmax layer to the target language and employed adaptation. Adaptation included one epoch of retraining the new network using only target language data; the learning rate for the adaptation was 0.001. Table 5.4 also includes the monolingual results obtained with DNNs trained on FBANKs and SGMMs trained on MFCCs. The number of hidden layers and neurons for each language was tuned using the Dev set based on the amount of training data.

To derive ISA features, two data selection methods were compared: random and entropy-based. First, to make the normalized graph Laplacian, 2k data points were randomly selected from each of the training languages and the similarity graph was made as explained in Section 5.2. In the second scenario, the selected random data from the first scenario was used as the initial set to apply the entropy-based data selection. The Parzen window size was jointly optimized with the graph hyperparameters for each language and we observed that $\rho = 100\rho_{sil}$ was a good choice for all languages where ρ_{sil} is obtained by Silverman's rule as explained in Section 4.4. σ, ξ were set to 300 and 1 respectively; these two hyperparameters were set empirically as we observed them to be fairly good choices for all languages.

In general multilingually trained DNNs with FBANK work better than monolingual systems; this suggests that even the full training data can be deemed low resource when we have more data available. Comparing the multilingual recognition results in Table 5.4 reveals that ISA with random data selection performs on par (sometimes worse, sometimes better) with the traditional FBANK features, whereas in Table 5.3, even with random data selection, ISA outperforms FBANK. On the contrary, ISA with entropy-based data selection brings consistent, though modest gains. We may note that in the multilingual scenario we only selected 2k data points from each language. This is a very small amount of data and probably not enough to represent the whole data structure for one language. Depending on the available memory, one might select more data for possibly a more reliable graph; however, in this work we aim at deploying an efficient use of ISA and therefore we proposed to

Table 5.4: WER(%) results for monolingual and multilingual systems. Results also compare ISA with/without data selection with the conventional FBANK features to the multilingual DNN.

Target language		Monolingual		Multilingual DNN (+adaptation)		
				FBANK	ISA	
		SGMM (MFCC)	DNN (FBANK)		Random	Entropy
FR	Dev	27.12	26.45	23.76	23.91	23.31
	Eval	24.35	23.90	22.45	22.52	22.16
SP	Dev	18.23	17.78	15.86	15.84	15.35
	Eval	10.68	10.42	8.87	8.78	8.43
PO	Dev	20.56	19.70	17.84	17.88	17.34
	Eval	21.43	20.97	18.42	18.58	18.05
GE	Dev	12.05	11.85	10.42	10.13	9.94
	Eval	20.0	19.49	16.08	16.00	15.82
RU	Dev	33.32	32.88	30.70	30.56	30.49
	Eval	31.98	31.37	28.93	28.68	28.65
MAN	Dev	19.37	18.51	18.14	18.50	18.02
	Eval	27.36	25.60	24.80	25.12	24.70

use the entropy-based data selection technique.

Although the improvements are modest, this is a remarkable result since it confirms that multilingual DNN training can be further improved by exploiting structural information of the data. Moreover, we obtained the ISA features with some approximations and we believe a more accurate manifold learning may lead to further improvements. For example, since a neural network is a universal approximator, one might argue that manifold learning can be accomplished with a DNN given a proper optimization criterion. In literature, [75] utilized DNN training to learn a function for graph embedding and [152] proposed to employ manifold learning as a regularization in DNN training. This opens up an interesting possibility to merge manifold learning and acoustic modeling in the same framework. While there are some key challenges, for example in term of complexity as pointed out in [152], designing a unified structure for DNN training and manifold learning in a multilingual setting is an interesting research direction.

5.7 Conclusion

We have proposed to use intrinsic spectral analysis as a manifold learning technique to extract a representation for speech which exhibits more language independence compared to the conventional FBANK and MFCC features as it has strong correlates with articulatory space. Furthermore, we showed that DNN bottleneck features trained on ISA features exhibit a high degree of node specificity for phonetic features and that these features are quite language independent. All these basic properties are in support of the claim that ISA features exhibit overall better language independent behavior than spectral features.

This was further confirmed in a set of crosslingual and multilingual ASR experiments. In low resource scenarios the advantage of ISA features was outspoken. In large vocabulary dictation tasks using multilingually trained DNN acoustic models, the gains obtained from the ISA feature representation were more modest, but were consistent across all languages. In the latter case a careful entropy-based data selection procedure per language was required to keep the size of the ISA data matrix within limits.

Chapter 6

Low Rank Multilingual DNNs for Improved Adaptation to a Low Resource Target Language

This chapter is adapted from the following article(s):

- Reza Sahraeian and Dirk Van Compernolle. A study of rank-constrained multilingual DNNs for low-resource ASR. In proceeding of ICASSP, pages 5420-5424, Shanghai, China, March 2016.
- Reza Sahraeian and Dirk Van Compernolle. Exploiting sequential low rank factorization for multilingual DNNs. In proceeding of ICASSP, pages 5025-5029, New Orleans, USA, March 2017.

6.1 Introduction

In this chapter we try to improve on a standard multilingual DNN for a low resource target language. The key idea is to reduce the number of parameters that need to be trained or updated by the low resource language data. In a multilingual DNN, the output layer is language dependent; thus we first focus on reducing the size of this layer. Furthermore, for a specific target language, it has been shown that additional gain over the purely multilingual DNN can be obtained by adapting the multilingual DNN using data from the target language [55]. Adaptation consists in retraining the multilingual DNN with target language data; hence, the adaptation performance depends on the size of the multilingual DNN vs the amount of adaptation data. Multilingual DNNs are usually huge and parametric and for a low resource language we have relatively small amount of data; hence, improving on the adaptation performance may be accomplished by reducing the total number of parameters in the multilingual DNN, possibly by loosing a bit in the multilingual DNN, but gaining more during adaptation.

The trend in DNNs is to increase the number of parameters to fully exploit ever-increasing training data [22]. This, however, entails long training time and slow prediction; and furthermore, once DNNs are being deployed on mobiles and embedded devices with little memory, such large models cannot be stored. This dilemma motivates several studies recently to investigate DNN size reduction without hurting the performance. In [173], it is shown that a large portion of the weight parameters in a DNN are very small and have a negligible effect on the output values of each layer which exploits the sparseness in DNN. Other techniques have been proposed which change the DNN architecture; for example, [176] proposed to shrink the hidden layers gradually from bottom to upper layers. Another work proposed to sparsify the weight matrices by employing regularizers [24]. [19] proposed a parameter sharing scheme using a hash function and a drastic reduction in model size was achieved. Moreover, a popular approach in this area is low rank matrix factorization [34, 127, 169]; the core idea is to represent the weight matrix as a low rank product of two smaller matrices. Low rank factorization (LRF) can be employed either with a linear bottleneck [127] or singular value decomposition (SVD) [169] to reduce the model size and training time while recognition accuracy is not significantly affected.

The main intent of the aforementioned studies is to reduce the model size and accelerate the DNN training and test time while no significant improvement is achieved. We show that the model size reduction via LRF in a multilingual DNN provides significant gain over a conventional multilingual DNN; first, only the final weight layer is factorized. Since the output weight layer needs to be trained

with language specific data, reducing the number of parameters is beneficial for under-resourced languages. We also discuss and compare other configurations of multilingual DNN training that reduce the number of parameters in the output layer: i) reducing the size of the last hidden layer by imposing a nonlinear bottleneck constraint; ii) employing the maxout strategy for the last hidden layer. We further extend the use of LRF for multilingual DNN by exploring several scenarios in which not only the final weight layer, but also other weight layers are factorized. We demonstrate that a considerable improvement is achieved in the adaptation phase by compressing the whole network as a much smaller number of parameters needs to be adapted to a specific target language with relatively small amount of data.

Moreover, LRF may associate with some problems: since factorizing the whole network may drastically degrade the performance, we suggest to employ the LRF layer wise and after each layer is factorized retraining is employed. This strategy also makes more sense if LRF is viewed as a regularization scheme.

6.2 LRF for multilingual DNNs

In the case of low resource ASR using multilingual DNN, LRF is particularly attractive as it reduces the number of independent parameters that should be estimated or adapted with low resource language data.

6.2.1 LRF on the softmax layer

In the multilingual DNN training, the parameters in the softmax layer are trained on the language specific data and reducing the number of parameters in this layer might be helpful in low resource settings. Let us denote the final weight matrix for language s by $\mathcal{A}^s$ with dimensions $n_H \times n_T^s$ where n_H is the number of units in the last shared hidden layer and n_T^s is the number of output targets for language s. Note that when a common output target using a universal phone set is used, there is only one output weight layer. In both scenarios, if there is a rank n_r for the final weight matrix, then there exists a factorization $\mathcal{A}^s = B^s \times C^s$ where B^s and C^s are full rank matrices of size $n_H \times n_r$ and $n_r \times n_T^s$ respectively. Now, in a multilingual low resource scenario we may want to further reduce the number of language dependent parameters by incorporating the matrix B^s in the layers that are shared across languages and thus $B^s = B$ for all languages as shown in Figure 6.1. Then, for a language s', we only need to train an output weight matrix of dimensions $n_r \times n_T^{s'}$, which is much smaller than $n_H \times n_T^{s'}$. In the case of multilingual DNN with separate

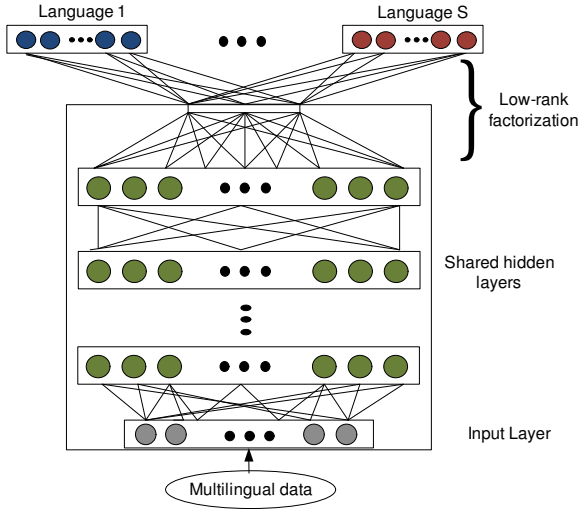


Figure 6.1: Multilingual DNN training with LRF in the final weight layer.

output layer per language, B^s is actually different for various languages and what we do is simply adopting the one corresponding to the target language and replace the rest with it. It is worth noting that in this approach there exists one extra weight layer in the shared components compared to the typical multilingual DNN; however, we show in the experiments that this is not very relevant.

LRF can be applied in two ways: first, by using SVD in which a $n_H \times n_T^s$ weight matrix layer $\mathcal{A}^s$ is decomposed as:

$$\mathcal{A}^s_{n_H \times n_T^s} \approx U_{n_H \times n_r} \Upsilon_{n_r \times n_r} V^T_{n_T^s \times n_r} \tag{6.1}$$

Then, we consider $B = U_{n_H \times n_r}$ and $C^s = \Upsilon_{n_r \times n_r} V^T_{n_T^s \times n_r}$ and replace $\mathcal{A}^s$ with these two smaller matrices [1] as described in [169]. Another approach is to restructure the network by replacing the weight matrix of the final softmax layer by two matrices with a linear bottleneck (LB) layer in between. Consider $\mathbf{h}^L$ as the input for the last weight layer and $\mathbf{z}^L$ as its output as presented in equation (2.11). There are two linear components in a row from $\mathbf{h}^L$ to $\mathbf{z}^L$ with a bottleneck layer in between. The propagation through a linear bottleneck can be expressed as:

[1] With SVD decomposition, the eigenvectors corresponding to the highest eigenvalues are kept.

$$\begin{aligned}\mathbf{z}^L &= \mathcal{A}^L[\mathcal{A}^{L-1}\mathbf{h}^L + \mathcal{B}^{L-1}] + \mathcal{B}^L \\ &= [\mathcal{A}^L\mathcal{A}^{L-1}]\mathbf{h}^L + [\mathcal{A}^L\mathcal{B}^{L-1} + \mathcal{B}^L]\end{aligned} \tag{6.2}$$

which is similar to the propagation through a single layer, where the weight matrix has limited rank. Therefore, we can configure the DNN with a linear bottleneck and the factorization is implicitly learned during DNN training.

Both of the above methods have been successfully used to compress DNNs while each comes with some upsides and downsides. For example, when we use a linear bottleneck, since the number of the parameters of the DNN is reduced, the overall training time can be reduced as well. The downside of this method, however, is that the bottleneck dimension has to be defined beforehand and for a new dimensionality we need to train a new DNN. On the other hand, we can train the conventional DNN and later apply SVD to factorize the weight layer; so, n_r can be tuned with less computational complexity. However, there is no impact on the multilingual DNN training time. More importantly, by factorizing with SVD we add some noise to the network which degrades the performance and we usually require a recovery phase to regain part or most of the lost information.

Moreover, we should note that LRF is not the only possible approach to reduce the number of parameters in the output layer; we investigate two other ways to do so:

- **Nonlinear bottleneck**: One possible way is to make a bottleneck constraint on the last shared hidden layer. This structure can also be motivated from [176] where DNNs with shrinking hidden layers are introduced based on the fact that as we increase the number of hidden layers and hidden units the weights in neural networks become more sparse. We reduce the dimensionality of the last nonlinear hidden layer from n_H to a new value n_R. Thus, the total number of weights which should be trained with low resource language specific data decreases to $n_R \times n_T$.
- **Maxout structure**: As presented in Table 2.1, maxout and generalized maxout (g-maxout) activation functions can reduce the dimensionality in the hidden layers. Therefore, we can apply the maxout or g-maxout nonlinearity in the last shared hidden layer to reduce the size of n_H. In this case, the number of parameters that need to be trained with low resource language data is decreased to $(n_H/G) \times n_T$; where G is the group size (Section 2.2.3).

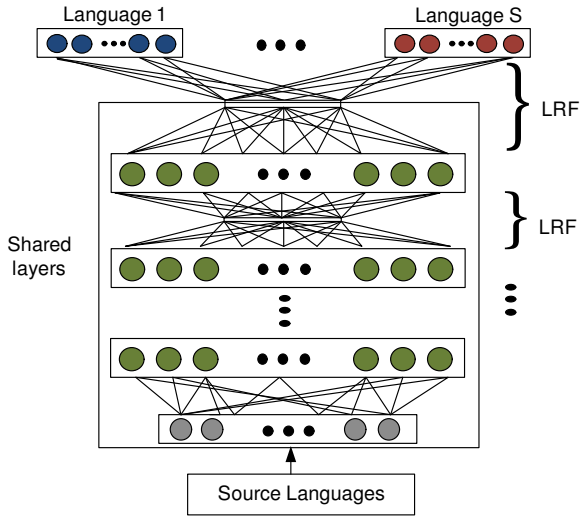


Figure 6.2: Multilingual DNN with LRF for all layers except the input layer.

6.2.2 LRF on all hidden layers

Next, we propose to extend LRF to other weight layers which leads to a huge reduction of the number of parameters in the multilingual DNN system. For the hidden layers, factorization is more straightforward and we simply need to factorize the weight matrix of size $n_H \times n_H$ into smaller matrices of size $n_H \times n_r$ and $n_r \times n_H$ so that $(n_H \times n_r + n_r \times n_H) < n_H \times n_H$. It is also worth noting that in our LRF approach we skip factorizing the input weight layer as it is a very small matrix compared to the other weight matrices (Figure 6.2). The LRF is applied after initial training of the multilingual DNN and before adaptation to a target language. In what follows, we employ SVD to factorize the whole network.

Factorizing all layers normally results in a huge model size reduction and consequently the multilingual DNN moves away from its optimal trained state. To remedy this problem, as suggested in [169], we need to retrain the whole network. However, depending on how far the factorized model has moved from the original DNN, we may require a long retraining process to gain back the lost information as much as possible. Moreover, the performance drop in the factorized DNN might never be completely recouped with retraining if the model is too far from the original one. Thus, it is of great interest to investigate if the possible gain by adaptation can make up for this drop.

6.2.3 Sequential LRF

Factorizing all layers together can drastically move the network away from the local optimum that was reached during training. This necessitates a recovery stage which can be simply a retraining. However, if LRF leads to a huge reduction in the number of parameters, a large amount of noise is added to the model and consequently a long retraining process is required. Not only is this not practically efficient, but also the factorized model might not be recovered if there is a huge drop in the size and performance. The second issue is that LRF of DNNs in the previous studies is mostly viewed as a way to reduce the model size; however, if being employed properly, it can take a role as regularizer and improve the model performance with respect to generalization. Nonetheless, we show in the experiments that factorizing all layers together causes a damage too much and is not a good regularization method.

Being motivated by the aforementioned issues, we propose to apply LRF sequentially to the weight layers. The reasoning is that we believe factorization of only one layer adds only a small amount of noise to the network and does not drastically move the model from its current state. Therefore, the chance to quickly retrieve the lost information increases. In terms of regularization, adding small noise sequentially is a more appropriate methodology than imposing a large amount of noise to the model in a one-shot manner.

For implementation, we start from the output weight layer; once this layer is factorized, we retrain the model with the whole multilingual data for a small number of iterations. Then, the layer below the output one is factorized and again the whole model is retrained briefly. This procedure continues until all layers are factorized (except the first input layer). If there are n samples in the multilingual training data pool which are shuffled, we can manage to retrain the DNN after each step of factorization using a subset of the training samples. For example, if there are 8 weight layers to be factorized and we retrain with $n/8$ samples after applying SVD to each layer, the whole factorization will be finished after 1 epoch. The important question which arises is how long each retraining stage needs to be. To answer this question, we investigated several scenarios in the experiments and we successfully show that even in one epoch, the whole factorization may be succeeded.

Table 6.1: Comparing PERs(%) for Afrikaans using monolingual and multilingual baseline systems.

Systems		Afrikaans data		
		1hr	5hr	10.7hr
Monolingual	HMM/GMM	23.09	16.87	14.81
	HMM/DNN	23.56	15.20	12.06
Multilingual DNN	Not adapted	18.66	12.83	10.89
	Adapted	18.52	12.64	10.79

6.3 Experiment 1: Compressing the Output Layer

6.3.1 Baseline results

This set of experiments is based on the Flemish-Afrikaans setting. The monolingual HMM/GMM system is similar to the one used in Section 4.6.2. The setting for the HMM/DNN system is also similar except that we used the ReLU nonlinearity instead of the 2-norm. The monolingual reference experiments yielded 505, 1380 and 2281 context-dependent states for 1hr, 5hr and 10.7hr training data respectively. The number of hidden layers and neurons per layer were tuned; the optimal number of hidden layers were 2, 3, 4, and the number of hidden units in each layer were 200, 400 and 500 for the respective settings.

The multilingual DNN is trained using the knowledge based phone mapping explained in Section 3.2.2. The output target included 4131, 4778 and 5422 tied-states for the multilingual HMM/GMM systems with 1hr, 5hr and 10hr Afrikaans included respectively. Furthermore, adaptation of the full DNN to Afrikaans data only was performed. The optimal number of hidden layers in each setting is 7, 8, and 8 for 1hr, 5hr, and 10.7hr Afrikaans respectively with the number of hidden units per layer equal to 1000. Table 6.1 presents the monolingual and multilingual baseline results for different settings of Afrikaans.

6.3.2 LRF results

In this section we examine the effectiveness of applying LRF to the last weight layer of the multilingual DNN as is shown in Figure 6.2. In other words, the final layer weight matrix, which has the size of $1000 \times n_T$ in our baseline multilingual DNN, is replaced with two matrices, one of size $1000 \times n_r$ and one of size $n_r \times n_T$ as explained in Section 6.2.1, this can be accomplished by configuring the multilingual DNN with a linear bottleneck (LB) or we can apply SVD

Table 6.2: PERs(%) for different low rank value using LRF of the softmax layer with both LB and SVD in the multilingual DNN for the Flemish-Afrikaans setting.

Afrikaans data	Methods	n_r		
		100	200	500
1hr	SVD	18.69	17.76	17.47
	LB	17.42	**17.34**	17.52
5h	SVD	12.68	**12.30**	**12.29**
	LB	12.51	12.34	13.36
10.7h	SVD	10.36	**10.29**	10.44
	LB	10.95	10.42	10.92

to the last weight layer of the conventional multilingual DNN; we investigate both. When SVD is applied, it is always necessary to fine tune the network; in this work, we retrained the multilingual DNN with multilingual data for 5 epochs after factorizing its last weight layer using SVD. Then, all the hidden layers together with the first weight matrix of size $1000 \times n_r$ are transferred with further adaptation to bootstrap the acoustic modeling for the Afrikaans language. The PERs for different choices of n_r are shown in Table 6.2.

Table 6.2 reveals further trends: first, using low rank decomposition of the multilingual DNN improves the performance compared to the conventional multilingual DNNs. Moreover, the PER reduction is more pronounced when the target language is more under-resourced. Also, it can be seen that although both SVD and LB approaches improve the performances if n_r is chosen properly, they do not yield the same results. This is possibly because of different initializations; in the LB approach, the two matrices are initialized together with other layers and trained discriminatively in the DNN training process while SVD factorizes the last weight layer of the already trained multilingual DNN and then the whole network is retrained.

However, reasonable questions that arise are how the nonlinear bottleneck would perform? and the low rank network has an extra weight layer compared to the multilingual baseline system so is the obtained improvement because of this extra layer? To answer these questions, we consider a scenario with 1hr Afrikaans training data; the simplest approach is to train a conventional multilingual DNN with 8 hidden layers where a nonlinear bottleneck constraint is applied on the last layer to have the width of n_r. Thus, the total number of weights in this multilingual system is the same as the 7-layer multilingual DNN with LRF of the last weight layer. The PER obtained for this system when $n_R = 500$ is 18.13% which is higher than the corresponding PER presented in

Table 6.3: PERs(%) for different choices of nonlinear bottleneck dimensionality (n_R) in the last layer of multilingual DNN for the Flemish-Afrikaans setting.

Bottleneck dim (n_R)	Afrikaans data			parameter reduction(%)
	1hr	5hr	10.7hr	
100	18.50	13.07	11.13	90%
200	18.29	12.92	10.94	80%
500	**18.13**	**12.37**	11.11	50%

Table 6.2.

As mentioned in Section 6.2.1, LRF is not the only possible way to reduce the number of parameters in the last weight layer. We examine two other possible approaches: one is by employing a nonlinear bottleneck layer and the other one is using g-maxout nonlinearity. In the former, a nonlinear bottleneck constraint is applied on the last shared hidden layer. Table 6.3 shows the PERs for different choices of the width, n_R, and percentage reduction in language dependent parameters compared to the baseline DNN system. Table 6.3 shows that the nonlinear bottleneck results in improvements over the conventional multilingual DNN system when 1hr and 5hr of training data is used for Afrikaans. However, for more available training data, e.g. 10.7hr, of the target language no improvement is observed. This can be due to the fact that there exists enough data to reliably estimate more parameters.

Next, we employ the g-maxout technique to reduce the dimensionality of the last shared hidden layers. To this end, all shared hidden layers have the ReLU nonlinearity but the last one has g-maxout nonlinearity. Table 6.4 shows the PERs for different choices of the group size, G. In our setting $n_H = 1000$ and therefore the effective output dimensionality of the last hidden layer equals to n_H/G. It is worth mentioning that we also investigated the conventional maxout technique and we observed that the performance of g-maxout was always slightly better. From Table 6.4 we can observe improved performance with a group size of 2; however, by increasing the group size the error rates go up again. Moreover, the observations suggest that when the target language is more under-resourced the improvement is more pronounced. The percentage reduction in language dependent parameters for each scenario is also provided in Table 6.4.

Table 6.4: PERs(%) using g-maxout nonlinearity for different choices of G as the final hidden layer function in the mutltilingual DNN for Flemish-Afrikaans setting.

G	Afrikaans data			parameter
	1hr	5hr	10.7hr	reduction(%)
10	18.51	12.83	10.90	90%
5	18.01	12.54	10.86	80%
2	**17.85**	**12.31**	**10.26**	50%

6.4 Experiment 2: Compressing All Layers

6.4.1 Baseline results

In this set of experiments, we used LRF in a more multilingual environment using the GlobalPhone data. First we focus on German (GE) as the target language, and Spanish (SP), Portuguese (PO), Russian (RU) and French (FR) as the auxiliary languages.

First, we constructed baseline systems for the three training sets in a monolingual fashion using HMM/GMM and HMM/DNN acoustic modeling. The number of context-dependent triphone states were 700, 1200 and 3100 with an average of 4, 9 and 13 Gaussian components per state for 1hr, 5hr, and 14.85hr German training data respectively. These parameters were tuned on the development set. WER for both development (Dev.) and evaluation (Eval.) sets are presented in Table 6.5 for HMM/GMM systems as well as HMM/DNN ones. The optimal number of hidden layers were 4, 4, 5 and the number of hidden units in each layer were 50, 200, and 300 for 1hr, 5hr and 14.85hr of training data respectively.

Then, a multilingual DNN was trained with a dedicated softmax layer for each language while the hidden and input layers were shared. The number of target context-dependent states were set to 3100 for each auxiliary language and the number of hidden layers and units per layer were tuned. We used a DNN with 7 layers for the setting including 1hr of German data and 8 layers for the two other settings; the number of nodes was 1500 per layer in all DNNs. The performance of the multilingual systems with and without adaptation is presented in Table 6.5. It is observed that no improvement was obtained by adaptation when only 1hr or 5hr of German data was available. With more available German data, we can see that adaptation yields a small improvement. This is a typical behavior for a multilingual DNN with a large number of parameters.

Table 6.5: WER(%) for German using monolingual systems and multilingual DNN trained on FR, PO, SP, TU and GE.

Settings		Monolingual		Multilingual DNN	
		GMM	DNN	Not adapted	Adapted
1hr	Dev.	22.84	21.41	18.74	18.78
	Eval.	35.38	34.90	32.54	32.57
5hr	Dev.	15.70	13.40	12.74	12.76
	Eval.	24.41	22.93	22.13	22.04
14.85hr	Dev.	13.95	11.85	11.15	11.02
	Eval.	21.36	19.49	18.78	18.36

6.4.2 LRF for all layers

First, the final weight layer of the best multilingual DNN was factorized using SVD and afterwards the whole network was fine tuned with multilingual data. The WERs for $n_r = 500$ are presented in Table 6.6 for both Dev. and Eval. sets. We also tried other bottleneck dimensions like 700 and 200 and we observed that $n_r = 500$ is a reasonable choice. For the 1hr German data, the system improves for the Dev set by just doing LRF which is the same behavior we observed in Afrikaans-Flemish experiments (Table 6.2); we attribute this improvement to the fact that the number of parameters that should be estimated with under-resourced language specific data has decreased. Moreover, when adaptation was applied, we observed improvements in all settings. This is most likely due to the fact that DNN model size is reduced. For example, for the setting with 14.85hr German data the number of the parameters in the multilingual DNN is reduced by a factor of 1.13. Finally, we experimented with factorization of ALL hidden weight layers. To this end, we took the best model obtained from the previous experiment; since the last weight layer of this model was already factorized, SVD was applied only on the hidden weight layers and the input weight layer was kept intact. In our experiment, $n_H = 1500$ and thus n_r needs to be chosen such that $(1500 \times n_r + n_r \times 1500) < 1500 \times 1500$. We set $n_r = 500$; so the number of parameters in each hidden weight layer is reduced by a factor of 1.5; afterwards, the whole network is retrained with multilingual data for 5 epochs. Table 6.6 compares the WERs for different factorized models before and after adaptation. From Table 6.6 we observe that the factorization of all hidden weight layers initially degrades the performance when 1hr and 5hr of data is available for German. This is not surprising as by applying LRF, some noise is added to all weight matrices and hence the network has moved away from the local optimum that was reached during training. However, when adapting the network from this starting point we ultimately reach a significantly better performance. This can be understood by the reasoning that LRF has created

Table 6.6: Comparing WER(%) for German data using multilingual DNN with LRF on the final layer and all layers ($n_r = 500$).

Settings		LRF on the final layer		LRF on all layers	
		Not adapted	Adapted	Not adapted	Adapted
1hr	Dev.	18.49	18.33	20.14	16.86
	Eval.	32.67	32.19	35.17	29.72
5hr	Dev.	12.87	12.69	14.93	11.82
	Eval.	22.28	22.19	23.25	19.82
14.85hr	Dev.	11.10	10.82	11.04	10.27
	Eval.	18.53	17.99	17.74	16.79

a network with fewer, but more relevant parameters. In 14.85hr scenario, we observe that the lost information after LRF of all layers can be well retrieved by multilingual retraining due to the availability of enough target language training data. Moreover, further improvement is achieved by adaptation like the other two scenarios. It is also important to note that the choice of learning rate in the adaptation phase is crucial; in our work, it was set to 0.0001.

6.4.3 Sequential LRF

In this part of the experiments, our multilingual setting includes the same 5 languages and all of them are considered for evaluation. Table 6.7 compares the recognition performance of the baseline systems with the conventional LRF method applied on all layers with $n_r = 500$. The multilingual DNN is trained on all available training data from the languages and it consists of 8 layers and 1500 nodes per layers [2]. After factorization, the multilingual DNN is retrained for 5 epochs.

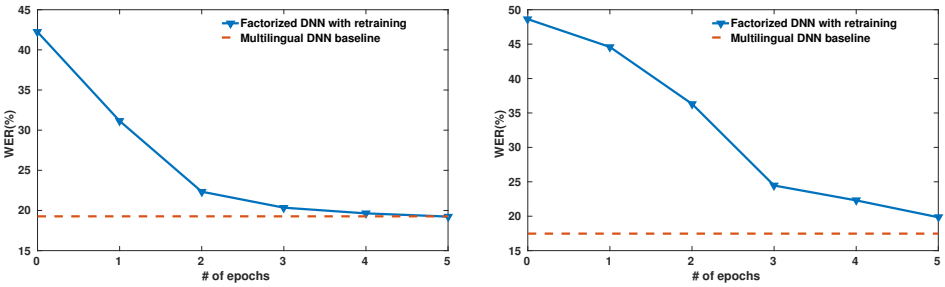
Figure 6.3 reveals how the performance of the multilingual DNN changes after factorization and during the course of retraining; the results in this figure are presented on Dev. sets for SP and PO as two examples. The following observations can be made from Figure 6.3. First, the LRF degraded the performance of multilingual DNN drastically which is not surprising as a big compression has happened. However, it can be seen that during the retraining process a big part of the gap was filled. After five epochs, the performance of the factorized multilingual DNN almost equals the original intact multilingual DNN for Portuguese; for Spanish, however, there still remains a small difference after 5 epochs. We could indeed increase the number of epochs and hopefully

[2]Noting that the multilingual setting is different from the one presented in Section 5.6.3 and accordingly the multilingual baseline results are different.

Table 6.7: Baseline results in WER(%) using monolingual and multilingual systems (with and without LRF) for the five languages: FR, SP, PO, RU and GE.

Settings		Monolingual DNN	Multilingual DNN		+ LRF	
			Not Adapted	Adapted	Not adapted	Adapted
FR	Dev.	26.45	25.90	25.15	26.04	25.07
	Eval.	23.90	23.65	23.48	24.25	23.04
SP	Dev.	17.78	17.47	17.14	19.85	16.37
	Eval.	10.42	9.73	9.57	10.16	9.14
PO	Dev.	19.70	19.27	18.87	19.24	18.38
	Eval.	20.97	20.48	19.84	19.75	19.47
RU	Dev.	32.88	32.64	31.99	34.42	31.32
	Eval.	31.37	30.60	30.25	33.26	29.96
GE	Dev.	11.85	11.15	11.02	11.04	10.27
	Eval.	19.49	18.78	18.36	17.74	16.79

better convergence would have been obtained for Spanish, but we should note that since retraining is applied multilingually, increasing the number of epochs might lead to overfitting for other languages. Besides, it is not efficient to have a long retraining because the choice of n_r is usually set empirically and we are interested in finding the proper value rapidly.



(a) Portuguese (b) Spanish

Figure 6.3: Tracking the WER(%) in retraining the low rank factorized multilingual DNN for SP and PO.

Next, we investigated the effectiveness and efficiency of the sequential LRF described in Section 6.2.3. Towards this goal, like the experiments for the conventional LRF, the original multilingual DNN without adaptation was deployed as the starting point. However, instead of factorizing all weight layers

Table 6.8: Comparing WER(%) on Dev. sets using conventional LRF and sequential LRF for different retraining durations in the multilingual setting with FR, SP, PO, RU and GE.

Target languages	Retraining duration (epochs) for sequential LRF					LRF
	1	2	3	4	5	
FR	25.90	24.67	24.57	24.54	24.44	26.04
SP	18.11	16.95	16.82	16.32	16.36	19.85
PO	20.08	19.14	18.99	19.01	18.80	19.24
RU	34.61	33.39	32.99	33.08	32.93	34.42
GE	11.67	10.89	10.51	10.52	10.70	11.04

at the same time, we applied the factorization layer by layer. Again, we set $n_r = 500$ for the sake of fair comparison to the previous experiments. At the beginning, only the final weight layer was factorized and the model was retrained for a fixed number of samples. Then, the next weight layer right before the final weight layer was factorized and again model was retrained for a while. This procedure continued until all hidden weight layers were factorized.

First, we examined different scenarios in terms of the retraining duration after each factorization. Since the multilingual DNN in our work includes eight hidden layers, factorization needs to be applied eight times (we don't factorize the input weight layer). After each factorization, the model is retrained for the specified number of training samples. Table 6.8 summarizes the results obtained from sequential LRF for different retraining duration on Dev. sets of different languages. In this table, "retraining duration" refers to the number of epochs required to have all layers factorized. The last column presents the results from Table 6.8 where we applied LRF to all layers and retrained the model for 5 epochs.

Interesting observations are made from Table 6.8. First of all, it can be seen that by applying sequential LRF, the proper compressed model can be obtained in a very short retraining duration. For example in Figure 6.3, we observed that five epochs of retraining with multilingual data was required to almost regain the lost information from LRF of all layers for Spanish; however, when LRF is employed sequentially in one epoch, we achieved the performance which is even closer to the original multilingual DNN. The results of applying sequential LRF in small number of epochs for different languages suggest that factorizing one individual layer led to a small reduction in model performance which was easily regained. The more interesting point being apparent in Table 6.8 is that in many cases sequential LRF even without adaptation improves the performance compared to the original multilingual DNN shown in Table 6.7. For example, the original multilingual DNN provides a WER of 17.47% on Dev. set for

Table 6.9: WER (%) for sequential LRF with and without adaptation for the multilingual DNN trained on FR, SP, PO, RU and GE. Relative WERs reduction compared to the standard multilingual DNN with adaptation are also presented.

Target languages		Sequential LRF in three epochs		Relative WER (%) reduction
		Not adapted	Adapted	
FR	Dev.	24.57	**24.01**	4.5
	Eval.	23.18	**22.64**	3.6
SP	Dev.	16.82	**15.94**	7
	Eval.	9.48	**9.03**	5.6
PO	Dev.	18.99	**18.00**	4.6
	Eval.	19.52	**18.53**	6.6
RU	Dev.	32.99	**30.52**	4.6
	Eval.	31.30	**29.00**	4.1
GE	Dev.	10.51	**10.10**	8.3
	Eval.	17.10	**16.44**	10.44

Spanish; taking this model as the starting point and applying sequential LRF in the duration of 4 epochs, the WER reduces to 16.32%. However, we can observe that when LRF was applied to all layers in one step, we could get the WER of 19.85% after retraining for 5 epochs. This improvement can be attributed to the fact that by factorizing only one weight layer a small amount of noise is added to the weight matrix and this can be viewed as a good generalization to the DNN. Besides, we should note that the gain by sequential LRF was achieved before adaptation and further improvement can still be obtained by adaptation of this compact model.

Moreover, we monitored the model performance prior and after each step of factorization. We observed that in many cases the reduction of WER after each step of factorization is less than 1% and retraining easily regained most of it. For example, for German data we observed that in the first step by factorizing only the final weight layer, the WER on Dev. set increased from 11.04% to 11.95%; or in the 6th step, the increase in WER was only 0.56%. This confirms our hypothesis that sequential LRF leads to only small drop in model performance at each step which can be easily retrieved.

Finally, we present the results of adaptation on top of the sequentially factorized multilingual DNN. Table 6.9 shows that further improvements were obtained by adaptation and the results highlighted in this table are the best recognition

performance we achieved. Comparing Table 6.7 with Table 6.9 reveals that sequential LRF provided a better factorized model compared to the conventional LRF, and consequently adaptation led to higher recognition performances. In the last column, we also present the relative WER reduction obtained by using sequential LRF and adaptation compared to the standard adapted multilingual DNN.

6.5 Conclusion

In this chapter, LRF of multilingual DNN was studied for improving low resource ASR. We examined different settings with different amount of data from target under-resourced languages. We demonstrated that LRF of the final weight layer gives a further improvement specially with adaptation and also if all hidden layers are factorized, a considerable improvement can be obtained during the adaptation. We also proposed to employ LRF in a sequential manner to deal with two major issues associated with the conventional LRF. In the experiments on five languages from the GlobalPhone dataset, we demonstrated the effectiveness of the sequential LRF. From the combined set of experiments we may draw the following conclusions: (i) in all scenarios, using conventional LRF together with adaptation improves the recognition results. (ii) The proposed sequential LRF significantly reduces the required retraining time and even with one epoch of retraining a proper compact model can be obtained. (iii) Sequential LRF in combination with adaptation can boost the results with 3.6-10.44% relative in comparison with the normal multilingual DNN with adaptation. Figure 6.4 provides an overview on the performance of low rank factorized multilingual DNN and its adaptation compared to the traditional multilingual DNN system.

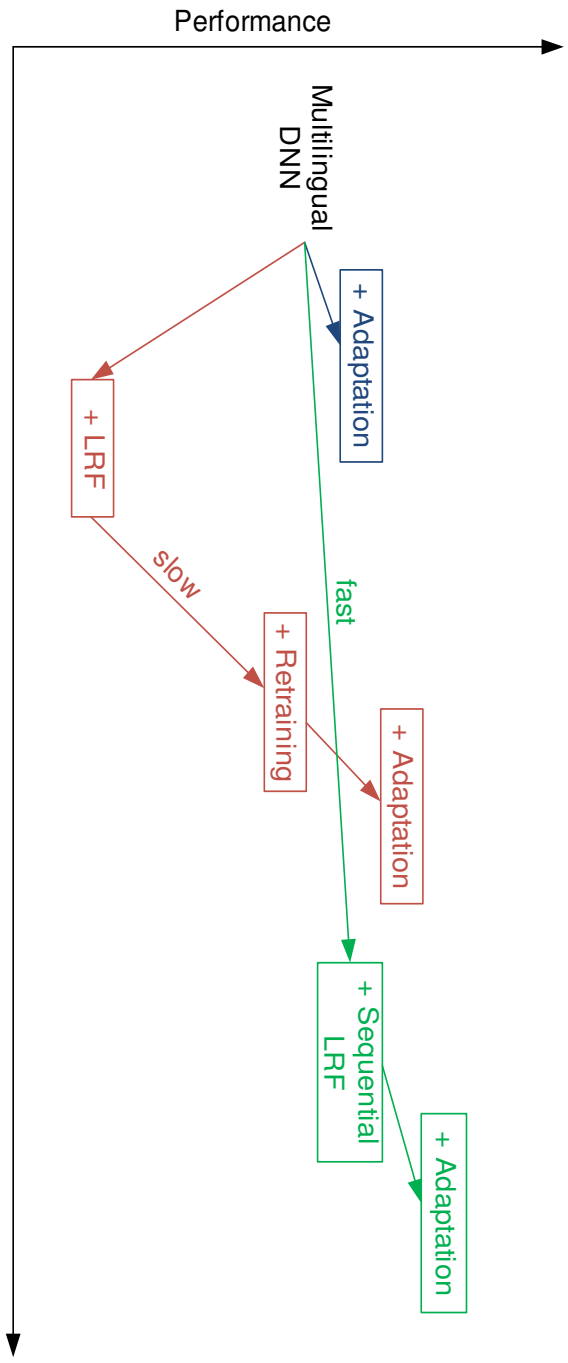


Figure 6.4: Overviewing the impact of LRF on a multilingual DNN with and without adaptation.

Chapter 7

Exploiting Similarities Between Source and Target Languages

This chapter is adapted from the following article(s):

- Reza Sahraeian and Dirk Van Compernolle. Using weighted model averaging in distributed multilingual DNNs to improve low resource ASR. In Workshop on Spoken Language Technology for Under-resourced Languages (SLTU), pages 152-158, Yogyakarta, Indonesia., May 2016.
- Reza Sahraeian and Dirk Van Compernolle. Cross-Entropy Training of DNN Ensemble Acoustic Models for Low Resource ASR. To be submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing.

7.1 Introduction

This chapter describes our work on multilingual DNN training where the main intent is to optimize the impact of individual source languages. The trend in multilingual DNN training is to exploit all available training data from various source languages whereas choosing only similar donor languages to the target language for multilingual and crosslingual acoustic modeling has shown to be effective. It creates a quandary; from one standpoint, the use of more data from various languages increases the chance of having a more generalized multilingual DNN with better context and phoneme coverage as well as allowing to train a bigger DNN. On the other hand, the mismatches between the target language and the source language(s) may obtrude by degrading the recognition performance.

The aforementioned dilemma motivated researchers in recent years to investigate how the similarity between source and target languages and the amount of training data influence the efficiency of multilingual DNNs. [83] showed that the correlation among source and target languages is of more importance when a modest amount of data from source languages is used; the conducted experiments in this work, however, are limited to one target language and a few source languages. In a richer experimental setup with BABEL languages and Fisher database, [53] shows that the neural network can still be over-weighted towards the source languages with large data. Furthermore, [99] provides sets of experiments to investigate the impact of using similar languages to a target language, and it shows that for the same amount of source language data, employing similar languages results in a better performance. Similar observations are made in [163] where the multilingual MLP trained on source languages from the same family as the target language improves the performance. These methods, however, entail training a new multilingual DNN from scratch for every new target language; a more subtle approach is proposed in [178] where a language identification (LID) scheme is deployed to pick the most similar languages to the target language from a given set of source languages and then use them to retrain a part of the network before adapting it to the target language. Moreover, [103] and [102] proposed to use a submodular data selection to choose a subset of multilingual data which is acoustically close to the target language and update the multilingual DNN for keyword search; the data selection in these works is carried out at the utterance level which is more elegant as not all data from a particular source language contributes equally to the multilingual DNN performance for the target language. This motivation prompted the authors in [21] to extend their LID approach to a frame level data selection.

Despite the insights obtained from the aforementioned studies, there still remain

some issues with respect to efficiency and accuracy. Firstly, data selection may require linguistic knowledge [99,163] or extra intermediate data driven techniques which in turn require heuristic thresholding [21, 103, 147]. Furthermore, if the size of the selected subset is too large, updating the multilingual DNN can take a long time and if the size is small, updating the hugely parametric multilingual DNN might not work well. To mitigate these problems, we propose two approaches based on weighted parameter averaging:

- First, we employ a weighted average modeling approach in the framework of distributed multilingual DNN training. Parallelized training of DNN acoustic models has been used in recent years in both monolingual [31, 112, 136] and multilingual [57, 96] scenarios to accelerate large network training. In our work, however, we utilize parallel training and model combination by proposing a weighted model averaging to improve the performance of a multilingual DNN. In this framework, different languages can have different effects on the multilingual DNN. Furthermore, we employ our method as a pre-adaptation technique in which a conventional multilingual DNN is retrained using the weighted model averaging approach and then the resultant model will be adapted using only target language data.
- Secondly, we propose a novel framework for multilingual DNN training which employs all the available training data and exploits the similarity between target and source languages at the same time; hence, avoiding the necessity for any explicit linguistic knowledge. Towards this goal, we borrow the idea of an ensemble with one *generalist* and many *specialists*. The generalist is derived from a multilingual DNN acoustic model trained on all available multilingual data; the specialists are the DNNs derived from the source languages individually. Then, the constituents in the ensemble are combined feraging scheme; the combination weights are trained to minimize the cross-entropy objective function. This framework implicitly guarantees that the final model performs better than or equal to the baseline as the multilingual baseline DNN exists in the ensemble. Moreover, unlike previous well-known system combination schemes, only one model is required during decoding.

7.2 Weighted Model Averaging in Distributed Multilingual DNNs

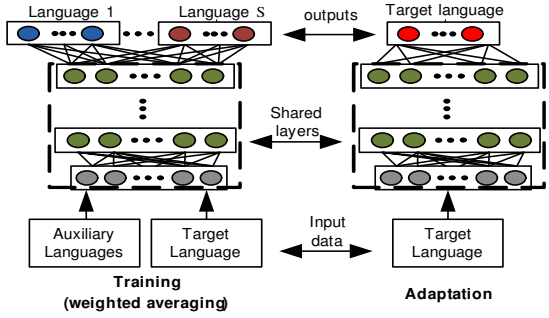
This section describes our first work where we investigate if assigning different weights to source languages benefits ASR for a specific low resource language.

These weights can be applied to the DNN parameters trained on the data of the source languages; however, in the conventional multilingual DNNs, the hidden layer parameters are trained with all source language data together. To overcome this problem, we exploit the distributed DNN training framework based on data parallelization [112]. This method allows multiple SGD being processed on different machines and the model parameters are averaged across all machines after a fixed number of samples has been processed. The averaged parameters are then redistributed for the next iteration and it will be repeated until all the data are processed for a specific number of epochs. The same framework can be utilized for multilingual DNN training. To this end, we use the language-based distributed learning algorithm in which each GPU uses the full data from one language and trains a normal DNN model. For the sake of efficiency, we consider the same number of samples to be processed for each language before averaging the parameters. This reduces the waiting time before the averaging, but we need to consider different epochs for languages depending on the amount of available training data for each language. Moreover, we only average the parameters of the input and hidden layers across languages and keep the output layers language dependent.

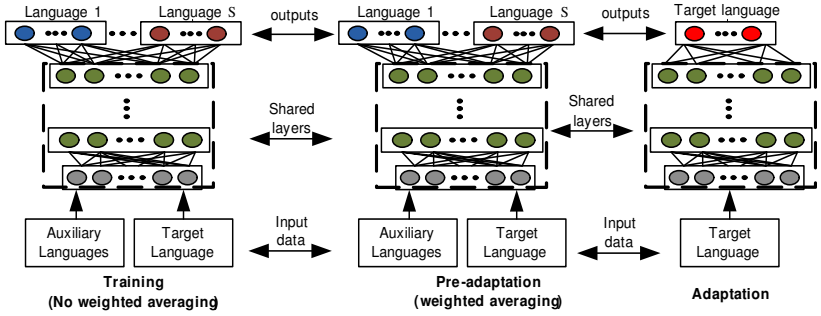
Using a weighted average approach allows us to control the effect of different languages on the multilingual DNN. Let's assume there are S languages being used for multilingual DNN training; then, we have S models being learned in parallel. The set of parameters in the hidden layers for the model j can be represented by Θ_j^{HL}; these parameters are supposed to be shared across all languages; thus, weighted averaging is applied on these parameters:

$$\Theta_{combined}^{HL} = \sum_{j=1}^{S} \lambda_j \Theta_j^{HL} \tag{7.1}$$

where λ_j is the corresponding weight for language j and $\sum_{j=1}^{S} \lambda_j = 1$ for $\lambda_j \geq 0$; this combination occurs periodically and the resultant parameters are redistributed as the starting point for further training. The key to successful application of this approach is to properly choose λ's for the training languages. For example, if data from the target language exists in the training set, it makes sense to give a higher weight to the parameters being trained over the target language data. This idea can be extended such that similar languages to the target language also take higher weights. Specifically in the case that only a small amount of target language data is available, the higher weights for the parameters being trained over matched languages may improve the acoustic model for the target language.



(a) Using weighted model averaging in training.



(b) Utilizing weighted model averaging as a pre-adaptation phase.

Figure 7.1: Two types of multilingual DNN training in which weighted model averaging can be employed in the training (a) or pre-adaptation (b).

The multilingual DNN parameters being trained in the weighted averaging framework (Figure 7.1 (a)), however, are shifted towards the languages with higher weights and thus hidden layers cannot be considered as language independent feature extractors. That is, for a new target language, a new multilingual DNN should be trained which is not of interest due to the slow DNN training procedure. To alleviate this problem, we also propose to use the weighted averaging in the adaptation phase where an intermediate retraining phase is applied after multilingual DNN training and before adaptation with target language data as shown in Figure 7.1 (b). This phase, which we call pre-adaptation phase, consists of retraining the multilingual DNN, which is already trained in a common way, for a small number of iterations using all languages in the weighted model averaging framework. This pre-adaptation can be helpful when we do not have a lot of target language data to traditionally adapt the multilingual DNN. It is clear that for a new target language, only the pre-adaptation and adaptation procedure need to be done which is much faster

than training a new multilingual DNN using the weighted model averaging from scratch.

Finally, we need to find the proper λ's; these weights can be assigned based on the acoustic similarities of the source languages to the target language. In this work, we first measure this acoustic similarity and then heuristically assign weights to the source languages.

7.3 Experiment 1: Weighted Model Averaging

This section includes our experiments to validate our first methodology explained in Section 7.2.

7.3.1 Setup

In this set of experiments, German (GE) was used as the target language, and Arabic (AR), Turkish (TU) and French (FR) as the auxiliary languages. We also examined a very low resource condition with only 1 hour of data from GE. The monolingual systems are the same as what we used in the previous chapter in Section 6.4. However, the multilingual DNN is different as the source languages are different.

7.3.2 Results

We investigate if some languages may play more important roles than others during the multilingual DNN training. We first set out a simple data driven approach to find the closeness between the auxiliary languages and the target language. To that end, we trained a Universal Background Model (UBM) with 400 Gaussian components using 39-dimensional MFCC features using German; given this UBM, the log-likelihoods of other languages data were calculated. Table 7.1 shows these values for the setting with 14.85hr of German data; it suggests that FR is the closest language to GE while AR is the furthest. We also observed the same order of closeness by using a UBM trained with 1hr German.

Then, we assessed the performance of our proposed weighted model averaging approach in the training phase as explained in Section 7.2. Throughout the experiments in this part, we considered the ratios of weights for various languages rather than assigning specific values to them. For example, the weight ratio set

Table 7.1: Averaged log-likelihood of data of FR, TU, AR and GE given the UBM trained on German data.

UBM	Averaged log-likelihood			
	GE	FR	TU	AR
GE	-90.69	-107.08	-108.09	-112.93

Table 7.2: WER(%) for German using different weight ratios in the the weighted model averaging approach for multilingual DNN trained on FR, TU, AR and GE.

Language weight ratio (FR:TU:AR:GE)	German data			
	1hr		14.85hr	
	Dev	Eval	Dev	Eval
(1:1:1:1)	18.91	33.77	11.26	18.28
(1:1:1:2)	18.62	33.66	11.10	18.04
(1:1:1:3)	18.72	33.53	11.07	17.87
(1:1:1:5)	18.87	33.82	11.00	17.74
(2:1:1:3)	18.91	34.12	11.09	18.07
(2:1:1:5)	18.95	34.23	11.08	18.07

of $(1:1:1:1)$ refers to a scenario that all languages take the same weight which is 0.25. First, we only adjusted the weight of the target language and assigned the same weights to the auxiliary languages as shown in the first four rows of the Table 7.2. The first row is the baseline system. The results reveal that giving a higher weight to the target language during the multilingual DNN training may improve the recognition performance. This is not surprising as the closest language to German is itself! However, we observe that in the case of having only 1hr of training data from German, the performance is degraded when the weight assigned to German is five times larger than other weights. We attribute this behavior to the fact that the DNN trained with such small amount of data is not reliable enough to take a very large weight. Thus, choosing a proper weight not only depends on the closeness of the corresponding language but also on the amount of training data available for that language.

Furthermore, we investigated scenarios where source languages also take different weights. The last two rows in Table 7.2 show the cases where the weight assigned to FR is bigger than AR and TU as FR shows the highest similarity to German based on Table 7.1. In the setting with 1hr German training data, no gain is obtained over the baseline multilingual system performance. In the other setting with 14.85hr German, improvements are obtained compared to the baseline multilingual system; however, we can observe that the recognition performance

Table 7.3: WER(%) for German using weighted model averaging as pre-adaptation for a multilingual DNN trained on FR, TU, AR and GE.

Language weight ratio (FR:TU:AR:GE)	German data			
	1hr		14.85hr	
	Dev	Eval	Dev	Eval
(1:1:1:2)	18.52	33.13	10.96	17.96
(1:1:1:3)	18.38	32.87	11.01	18.18
(1:1:1:5)	18.66	32.64	11.01	17.84
(2:1:1:5)	17.89	33.04	11.04	17.99

when FR has the same weight as TU and AR is the best. We also conducted some other experiments with different combination of weight ratios and the same observation was always made. It seems that in our setting giving different weights to the auxiliary languages during the training cannot benefit the target language's acoustic model.

Next, we used the weighted model averaging scheme in the adaptation phase. To this end, the conventional multilingual DNN was used as a starting point for retraining with the proposed method. Then, the resultant model was adapted with German data. Table 7.3 shows the WERs for some weight ratio combinations. The results show that the improvements achieved is comparable with those presented in Table 7.2. Another interesting trend is that in this set of experiments a small number of epochs, (one or two), was required for the pre-adaptation phase which makes it much more efficient and faster than using weighted model averaging in the training. Moreover, unlike what we observed in Table 7.2, in the case FR takes a higher weight than AR and TU the performances are improved in both settings. This confirms our hypothesis that giving higher weights to similar languages can be beneficial for target language acoustic modeling.

7.4 Cross Entropy Training of DNN Ensemble Acoustic Models

This section explains our second work where the main intent is to improve on multilingual DNN training for a low resource language by exploiting two types of knowledge: i) the generalization and phoneme context coverage as well as better representation gained by a large network ii) relevant knowledge from individual source languages. The former can be gained by a multilingual DNN trained on all available data from as many languages as possible. For the latter, we utilize a set

of DNNs derived from individual source languages; in other words, we form an ensemble of DNN acoustic models, and accordingly, our method can be viewed as an ensemble learning [36], as we aim to produce an accurate yet diverse ensemble of models and combine them to extract possible complementary information. The common trend in ensemble learning is to combine the constituent models during decoding which makes using an ensemble at test time expensive. To overcome this problem, it has been shown that the knowledge in an ensemble can be compressed into a single model for deployment [11, 62]. This strategy was also proposed with the concept of teacher-student nets to train a simple model (student) which mimics a complex model (teacher) [2]. In our framework, the teacher is a DNN ensemble and the student model is a single model which captures the useful information in the ensemble.

Our methodology to derive the single student model is motivated by the DNN parameter averaging scheme in [112] and our work explained in Section 7.2. The DNN acoustic models in the ensemble are combined by weighted parameter averaging; however, unlike our work in Section 7.2, the combination is done only once and we increase the flexibility by letting individual components of DNN models to be combined. Most importantly, in this work, we employ a learning scheme with CE objective function to train the combination weights; this guides the learning process in the same direction as the acoustic model training. This framework, however, requires all models in the ensemble to have the same structure; also, they should be trained with the same initialization. This is not trivial and hence we first address these issues and explain how to create such models efficiently, and then we describe the combination procedure.

Furthermore, it is worth mentioning that a successful approach to compress knowledge from a cumbersome teacher model to a simple student is called knowledge distillation which has recently become popular in the speech community [16, 17, 26, 45, 88], and the main idea is to train a student model which matches the soft output of the teacher. While the comparison between our combination scheme and knowledge distillation is beyond the scope of this work, for low resource ASR task, our combination technique is simpler in certain aspects. Unlike [17] and [26] which require one stage of combining predictions and then using the soft labels to train the student model, in our method, combination directly generates the student model. Moreover, when the student is trained on the soft labels (posteriors or logits), its performance depends on the availability of large amounts of data [2, 11] or even a separate transfer data set [62] which can be a problem as we want to derive a solution for low resource languages.

7.4.1 Language Dependence of Hidden Layers: a motivation for DNN ensemble

While the parameters in DNN hidden layers show a great degree of language independence, it is intuitively plausible that training or updating these parameters with enough data from the target language or even closely related languages may bring further improvement. In this section, we investigate the language dependence of hidden layers by viewing them as feature extractors. Towards this goal, we examine the outputs of different hidden layers in the DNN. The main idea is to investigate how this representation varies with respect to the source language used to train the hidden layers and also across different hidden layers.

To inspect the transformed features, we look at their representational properties with respect to the phonemic categories. Looking at phonemic categories instead of phonemes is mainly for simplification; moreover, it is shown in [101] that interpreting the output of hidden layers units based on phonemic categories is sensible. In this work, 8 categories based on manner and place of articulation are constructed: front-vowel, back-vowel, open-vowel, plosive, labial, nasal, fricative, and approximant. For quantitative interpretation, the F-ratio is utilized as the ratio of between-group variability to within-group variability. We consider SP(1hr) as the target language and use the training portion of the five source languages (PO, AR, SW, MAN, GE) to form different crosslingual settings. For the sake of comparison, we also consider a monolingual setting where the DNN is trained on the full training data from SP. For each language we train a DNN with 5 hidden layers and 300 units per layer, and the number of target context-dependent states was set to 3100. Figure 7.2 compares the F-ratios obtained from the outputs of different hidden layers from the DNNs.

Interesting observations can be made from Figure 7.2; first of all, it is apparent that the source languages impact the hidden layers differently. This is in line with the findings of previous works that the choice of source languages matters; also, it clearly shows that the hidden layers are not totally language independent. Furthermore, going from the first layer to the last one, the F-ratio trends upward and then downward in all crosslingual settings while this is not the case for the monolingual one. This can be attributed to the fact that the top layers are expected to be more language dependent. More interestingly, we can observe that the influence of the source languages varies among different layers; for example, for the first three hidden layers SW gives higher F-ratio than PO and for the last two ones PO outperforms all others. This observation gives the intuition that if we want to exploit maximum information from the source languages individually, it is probably better to look at the individual layers rather than the DNN as a whole.

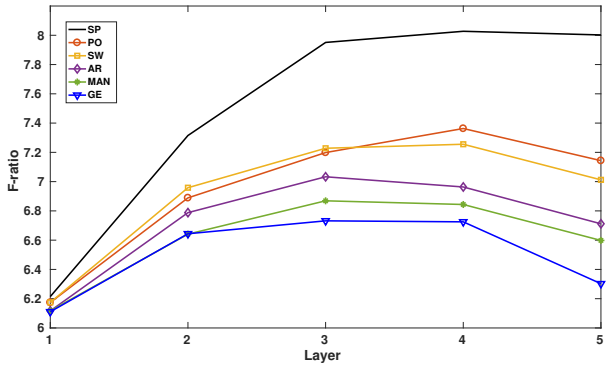


Figure 7.2: Comparing F-ratio for 8 phonemic categories in SP(1hr) from the output of the hidden layers of DNNs trained on various source languages.

The aforementioned observations can be insightful if we want to extract information from individual source languages.

7.4.2 Employing Ensemble of Deep Nets to train multilingual DNN

This section describes our proposed DNN-based multilingual teacher-student training algorithm. Let's assume that the linear component parameters in a normal DNN with L nonlinear components and $L+1$ linear layers are presented by $\Theta = \{\theta^0, ..., \theta^L\}$, where θ^l represents the set of whole parameters in the linear layer l. Our method consists of two stages: first, we construct an ensemble of generalist and specialist DNN acoustic models which plays the role of teacher, and then, the student model is obtained from a weighted combination of the DNNs in the ensemble.

Constructing the Ensemble of Deep Nets

Constructing a good teacher ensemble is a prerequisite to produce a well-performing student in a fast and efficient way. In this work, we use an ensemble of one "generalist" and many "specialists" as presented in [62]. Following we summarize how the ensemble is constructed:

1. The multilingual DNN is trained in a traditional way.

2. The hidden layers are reused and the out put layer is trained for the target language. This model is the generalist.

3. The multilingual DNN is adapted to each source language in a common way.

4. The hidden layers from the adapted DNNs in the previous step are reused and the output layer is trained for the target language. This gives us the specialist models.

The generalist is a DNN derived from a conventional multilingual DNN and its hidden layers are trained on multilingual data. Each specialist is derived from one specific source language so that the hidden layer parameters are shifted towards that source language in the acoustic space. To construct the specialists, the most elegant approach is to adapt the multilingual DNN to each source language in the traditional fashion, and then the hidden layers are deployed for target language. This setting has some major advantages: first, it provides a homogeneous ensemble in which models have the same type and architecture which is a prerequisite for our proposed combination scheme. Secondly, implementing this system is simple and easy to be parallelized. Moreover, the specialists are evolved from the generalist and weighted averaging of these models' parameters is justifiable.

Moreover, it is worth mentioning that when size of the training data from each source language is small relative to the huge multilingual DNN, the application of a subtle approach like LRF is important to ensure that adaptation performs well. Also, the generalist in the ensemble is actually the baseline model we use for the low resource target language; this implies that if specialists cannot provide relevant information, the student model can learn only from the generalist and consequently the student model will not perform worse than the baseline.

Combination of DNNs in the Ensemble

Once the ensemble of the specialists and the generalist is constructed, we train the student model with weighted averaging. The combination occurs for individual linear layers; thus, the hidden layer l in the student model is obtained by:

$$\theta^l_{student} = \sum_{j=0}^{S} g(\lambda^l_j)\theta^l_j \tag{7.2}$$

where λ's are the raw combination weights which should be found, and $g(.)$ is a normalized log-linear function to constrain the interpolation weights to be be positive and $\sum_{j=0}^{S} g(\lambda_j^l) = 1$:

$$g(\lambda_j^l) = \frac{\exp(\lambda_j^l)}{\sum_{i=0}^{l} \exp(\lambda_i^l)} \tag{7.3}$$

Given $L+1$ linear layers and $S+1$ models in the ensemble, $(S+1) \times (L+1)$ weights need to be learned in aggregate. Let's assume the vector $\mathbf{z}^l$ corresponds to the pre-nonlinearity activation and $\mathbf{h}^l$ is the neuron vector at the lth hidden layer in the student model so that:

$$\mathbf{z}^l = \theta_{student}^l \mathbf{h}^l \quad and \quad \mathbf{h}^l = \phi(\mathbf{z}^{l-1}) \tag{7.4}$$

with ϕ as the activation function. The output targets in the student model and the constituent models in the ensemble are identical and obtained from a forced alignment by an HMM/GMM system trained on the low resource language training data. The output layer is the softmax function to estimate the class posterior probabilities:

$$\mathbf{P}(y_i|x_i, \boldsymbol{\lambda}) = \alpha_{y_i} = \frac{exp(z_{y_i}^L)}{\sum_{\mathtt{k}=1}^{\mathtt{K}} exp(z_{\mathtt{k}}^L)} \tag{7.5}$$

where $\boldsymbol{\lambda}$ refers to the set of all λs. y_i is the label for the frame x_i in the training data. $z_{\mathtt{k}}^L$ is the $\mathtt{k}$th element in vector $\mathbf{z}^L$ and $\mathtt{K}$ is the total number of classes at the output. We optimize the combination weights, $\boldsymbol{\lambda}$, jointly by using the cross-entropy objective function:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \mathbf{P}(y_i|x_i, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \mathcal{L}(x_i, y_i) \tag{7.6}$$

where N is the total number of samples in the training data.

For the output layer, the gradient is:

$$\frac{\partial \mathcal{L}}{\partial \lambda_j^L} = \frac{\partial \mathcal{L}}{\partial g(\lambda_j^L)} \frac{\partial g(\lambda_j^L)}{\partial \lambda_j^L} \tag{7.7}$$

the second derivative can be easily derived from equation (7.3); for the first one we have:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial g(\lambda_j^L)} &= \sum_{i=1}^{N} \frac{\partial \mathcal{L}(x_i, y_i)}{\partial g(\lambda_j^L)} = \sum_{i=1}^{N} \sum_{\mathtt{k}=1}^{\mathtt{K}} \frac{\partial \mathcal{L}(x_i, y_i)}{\partial z_{\mathtt{k}}^L(i)} \frac{\partial z_{\mathtt{k}}^L(i)}{\partial g(\lambda_j^L)} \\ &= \sum_{i=1}^{N} [\boldsymbol{\alpha}(i) - \mathbf{d}(i)] \circ [\theta_j^L \mathbf{h}^L(i)]\end{aligned} \tag{7.8}$$

$\mathbf{d}(i)$ is the binary vector of size $\mathtt{K}$ with all elements equal to zero except for element y_i. $\boldsymbol{\alpha}(i)$ is the vector of posteriors at time i and $\circ$ represents the dot product. For the layer $L-1$ we have:

$$\frac{\partial \mathcal{L}(x_i, y_i)}{\partial g(\lambda_j^{L-1})} = \sum_{n=1}^{n_H} \frac{\partial \mathcal{L}(x_i, y_i)}{\partial z_n^{L-1}(i)} \frac{\partial z_n^{L-1}(i)}{\partial g(\lambda_j^{L-1})} \tag{7.9}$$

n_H is the number of neurons in the hidden layers. We can extend the first term as follows:

$$\frac{\partial \mathcal{L}(x_i, y_i)}{\partial z_n^{L-1}(i)} = \sum_{\mathtt{k}=1}^{\mathtt{K}} \frac{\partial \mathcal{L}(x_i, y_i)}{\partial z_{\mathtt{k}}^L(i)} \frac{\partial z_{\mathtt{k}}^L(i)}{\partial h_n^L(i)} \frac{\partial h_n^L(i)}{\partial z_n^{L-1}(i)} \tag{7.10}$$

$\partial h_{n'}^L(i) / \partial z_{n'}^{L-1}(i)$ is the derivative of the nonlinear activation function; for ReLU it is 1.0. By plugging (7.10) in (7.9) and rewriting in the matrix format we have:

$$\frac{\partial \mathcal{L}(x_i, y_i)}{\partial g(\lambda_j^{L-1})} = [\theta_j^{L-1} \mathbf{h}^{L-1}(i)] \circ [(\sum_{j=0}^{S} g(\lambda_j^L) \theta_j^L) \times (\boldsymbol{\alpha}(i) - \mathbf{d}(i))] \tag{7.11}$$

$\times$ refers to the normal matrix multiplication. For the remaining gradients we follow the same strategy. Noting that extracting the gradients is very similar to what we do during DNN training except that the parameters that should be learned are the combination weights $\boldsymbol{\lambda}$, and we used L-BFGS [167] as the learning algorithm.

7.5 Experiment 2: Ensemble of DNNs

In this section we conduct a set of experiments to examine our method described in Section 7.4.

Table 7.4: Baseline monolingual and multilingual results in WER(%) for SP and RU as the target languages while the source languages are: GE, PO, MAN, AR, SW, FR and TU.

Target languages		SP			RU		
Setting		1hr	5hr	17.55hr	1hr	5hr	21.10hr
Monolingual	Dev.	44.11	25.30	17.78	51.70	38.47	32.88
	Eval.	26.97	12.85	10.42	47.30	36.46	31.37
Multilingual	Dev.	25.68	19.72	15.78	41.47	34.25	30.40
	Eval.	14.04	10.44	8.89	39.44	32.36	28.99

7.5.1 Setup

In this set of experiments, 9 languages were used from the GlobalPhone dataset; two of them for test and 7 languages were taken to form the set of source languages which includes GE, PO, MAN, AR, SW, FR and TU. For the test (target) languages, we chose SP and RU and examined different amounts of training data; to simulate low resource conditions, two subsets containing 1 hour (8 speakers) and 5 hours (40 speakers) of data are constructed, both using randomly selected 7-8 minutes of speech for each of the selected speakers, and we will denote them, e.g. in the case of Spanish, as SP(1hr) and SP(5hr).

7.5.2 Baseline results

First, we constructed baseline monolingual systems. We trained conventional 3-state left-to-right HMM triphone models using MFCC features, and then the alignments of the triphone systems were used to train monolingual DNNs. The number of target context-dependent states was set to 600, 1200 and 3100 for 1hr, 5hr and full training data respectively. For each setting, the size of the DNN was tuned using the development sets. The optimal number of hidden layers was 2, 3 and 5, and the number of hidden units in each layer was 100, 300 and 300 for 1hr, 5hr and full training data respectively; the monolingual results in WER for both development (Dev.) and evaluation (Eval.) sets are shown in Table 7.4.

Our multilingual baseline system is based on a multilingual DNN with sequential LRF 6.2.3. To construct this system, first a conventional multilingual DNN with 10 hidden layers and 1500 nodes per layer was trained using the full training data from the source languages. The hidden and input layers were shared while each language had a dedicated softmax layer and the number of target context-dependent states was set to 3100 for each language. Then, we applied

SVD on the weight layers one by one, and after each layer is factorized, the whole network is retrained with a part of multilingual data. In our experiment, the low rank value was empirically set to 300 and all hidden layers (except the input layer) were factorized during 4 epochs of retraining. The size of the factorized multilingual DNN was reduced by a factor of 2.5; yet, its performance was very close to the original intact multilingual DNN. Next, the hidden layers were reused and only the last weight layer was trained with the data of target language, and finally, the whole network was adapted with target language data for 1 epoch. The learning rate used for the adaptation was set to 0.0001. Table 7.4 presents the recognition performance of the baseline multilingual DNN.

7.5.3 Ensemble of DNNs results

As explained in Section 7.4.2, the next step in our methodology is to construct an ensemble of deep nets. Towards this goal, the factorized multilingual DNN was adapted to different source languages. These models together with the original factorized multilingual DNN are used as explained in Section 7.4.2 to construct the ensemble.

Prior to embarking on the combination part, it is of interest to observe the performance of individual base models in the ensemble. To this end, Table 7.5 presents the recognition results on Dev. sets using the specialists and the generalist. The performance of specialists depends on the amount of available training data for source languages and their acoustic similarities to the target language. For example, among all specialists the one corresponding to PO results in the best performance for SP(1hr) which could be expected as PO is known to be a similar language to Spanish. This, however, does not necessarily mean that other models do not contain relevant information. The idea in ensemble learning is to induce diversity among the models and benefit from their complementary information.

Moreover, it is observed that the generalist performs very well in all settings; this reaffirms the importance of having this informative model in the ensemble. Furthermore, the performance differences among constituent models in the ensemble are more highlighted in low resource conditions; this is sensible as with more data a better output layer is trained which can more likely make up for the hidden layer differences.

Next, we investigate the DNN combination method explained in Section 7.4.2. The constituent models in the ensemble are rank-constrained DNNs where each linear weight layer is replaced by two consecutive linear components. We could reconstruct them prior to the combination; however, since the compressed

Table 7.5: WER(%) of Dev. set for the base models in the ensemble (without adaptation). The target languages are RU and SP and the ensemble includes the models derived from GE, PO, MAN, AR, SW, FR and TU.

Target languages		SP			RU		
Setting		1hr	5hr	17.55hr	1hr	5hr	21.10hr
Specialists	AR	27.06	20.21	16.80	43.02	35.22	31.39
	MAN	27.16	20.49	16.92	43.02	35.70	31.56
	FR	26.60	20.40	16.79	42.85	35.20	31.28
	GE	27.11	20.44	16.73	42.55	35.17	31.44
	PO	25.93	19.83	16.40	41.88	34.58	**31.12**
	SW	26.79	20.39	16.38	42.62	35.35	31.59
	TU	26.57	19.95	16.49	42.32	34.94	31.55
Generalist		**25.71**	**19.72**	**16.33**	**41.47**	**34.48**	31.16

structure is more suitable for the adaptation phase, we combine the models with the low rank structure so that the student model will be compressed too. This does not change the combination scheme except that since we have more linear components, more combination weights need to be trained.

The combined model is initialized with uniform combination weights. Then, the L-BFGS method was employed to learn these weights to minimize the loss function. The combination weights were updated for 200 iterations. To have a better understanding of the role of generalist and specialists, we conducted two combination scenarios: first, only the specialists were combined and the generalist is left out; this combination system is called Comb1. Then, the generalist is also available during the combination; this system is denoted by Comb2. Table 7.6 presents the results for these two scenarios.

Table 7.6 reveals the following trends; first, in Comb1 system, the student model learned from the combination of the specialists outperforms the specialists individually; this suggests that complementary information exist among the specialists and were exploited during the combination. Moreover, in all settings Comb2 system results in a better recognition performance than Comb1 system; this reveals the importance of the generalist as well as the fact that further complementary information can be gained in the ensemble of generalist and specialists. Also, in all cases, the improvements obtained from the combination systems are more pronounced for the more under-resourced settings; this is sensible as with larger amount of data, the useful knowledge can be learned from within language data rather than the donor languages. Finally, the student model from Comb2 system was adapted to the target language data and the recognition results are presented in the Table 7.6. It can be observed that after

Table 7.6: WER(%) results for the student model derived from the combination of only specialists (Comb1), combination of specialists and generalist (Comb2), and also adaptation of Comb2 system. The target languages are RU and SP; the source languages are: GE, PO, MAN, AR, SW, FR and TU.

Combination systems		Target languages					
		SP			RU		
		1hr	5hr	17.55hr	1hr	5hr	21.10hr
Comb1	Dev.	25.51	19.59	16.35	41.23	34.51	31.13
	Eval.	13.85	10.70	9.29	38.98	33.02	29.75
Comb2	Dev.	25.31	19.60	16.26	41.12	34.37	31.03
	Eval.	13.82	10.69	9.12	38.82	32.90	29.65
Comb2 + adaptation	Dev.	25.20	19.50	15.47	40.52	33.71	30.30
	Eval	13.68	10.23	8.84	38.56	32.02	28.79
Relative WER reduction (%)		**2.22**	**1.56**	**1.17**	**2.26**	**1.31**	**0.5**

adaptation, the student model performance is always better than the baseline multilingual DNN system in Table 7.4; the last row presents the relative WER reduction compared to the multilingual baseline system. The improvements are consistent although being modest in the cases we use the full training data from the target languages. This is a compelling achievement as the student model can be obtained in a very fast process by learning a small number of combination weights. For example, Figure 7.3 depicts how the objective function changes during the course of training in the Comb2 system for RU(1h) and RU(5h) settings; it is observed that the convergence is obtained in less than 20 iterations.

Moreover, it is of interest to look at the learned combination weights. Figure 7.4 shows the values of the combination weights for SP(5hr) and RU(5hr) settings in the Comb2 system. The DNNs in the ensemble have 21 linear layers and thus 21 weights were trained for the combination. Some interesting observations can be made: in both cases we can observe that many of the top layers are only taken from the generalist; to construct the bottom layers, however, many of the constituents contribute. Moreover, it is apparent that the specialists contribute differently; in the SP(5hr) setting the specialist derived from PO plays an important role. For the RU(5hr) setting TU seems to be very important while MAN is almost left out. Figure 7.4 manifests how the combination scheme constructs the student model by exploiting information from the individual layers.

Finally, we briefly compare our combination technique with the combination

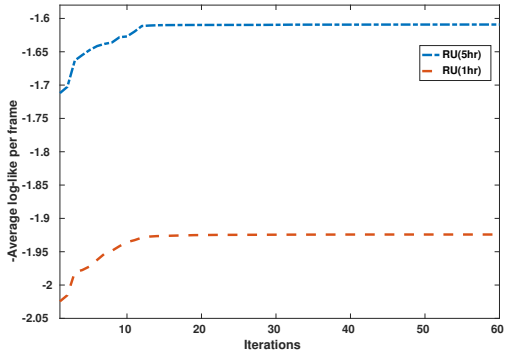


Figure 7.3: Tracking the objective function over the course of training the combination weights for two settings RU(1hr) and RU(5hr) in Comb2 system.

in the lattice level using minimum Bayes risk [168]. This is not the main idea of this work; however, it is of interest to compare our method with the combination of lattices as a popular combination methodology. To this end, we used the lattices created for the decoding by individual base models for SP(1hr) setting in Table 7.5. The lattices were then combined by weighted averaging. In this case we found the proper weights with a greedy search. A WER of 25.48% was obtained which is better than the performance of the constituent models in the ensemble (Table 7.5). However, it is worse than what we gained with our proposed combination scheme. In addition, we should note that the lattice combination system requires all DNN base models to be available for the decoding which is not efficient in terms of memory and timing; whereas, in our acoustic model combination framework, one single student model will be deployed in the decoding phase.

7.6 Conclusion

In this chapter we proposed two methodologies based on DNN parameter averaging to exploit relevant information from individual source languages. First we used the distributed DNN training framework in which DNNs were trained in parallel and being combined repeatedly. In this framework, we proposed to train each DNN on one specific source language and in the combination stage, different weights were assigned to the source languages. We observed that this strategy can improve the multilingual DNN performance for a low resource target language.

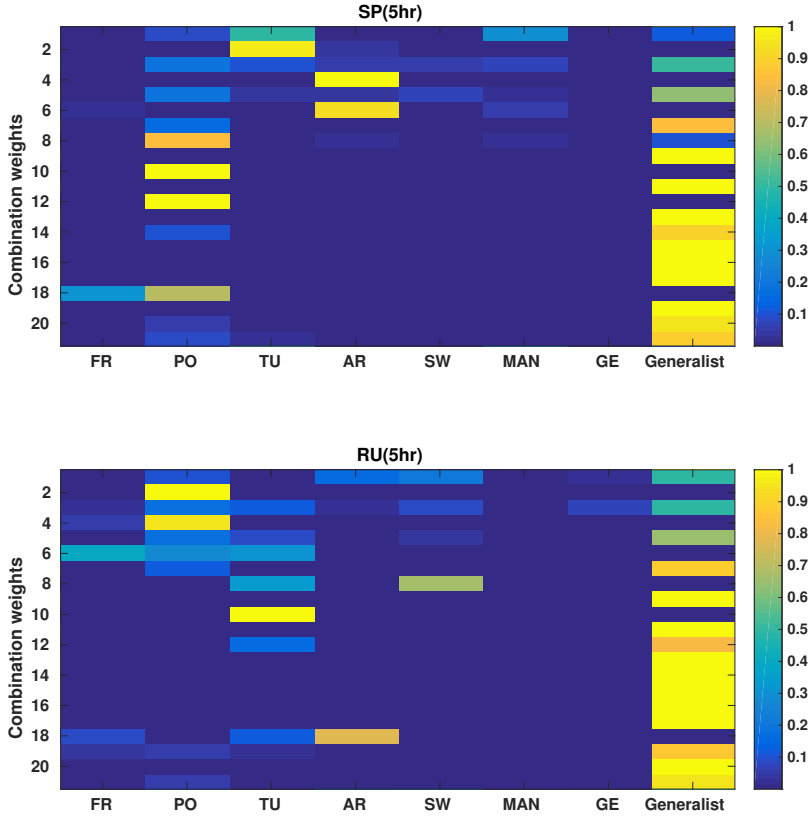


Figure 7.4: Combination weights obtained for setting SP(5hr) and RU(5hr) in Comb2 system.

Next, we proposed to construct an ensemble of DNN acoustic models including one generalist derived from the multilingual DNN and many specialist each being derived from one specific source language. By weighted averaging of the linear components of the base models, a single model is obtained which outperforms individual models in the ensemble. The combination process is fast and the combination weights are learned via the CE objective function.

Chapter 8

Conclusion

This chapter concludes the thesis by reviewing the original contributions of our work as well as providing several research directions for future work.

8.1 Original Contributions

This thesis centres around the problem of acoustic modeling in impoverished scenarios. We have taken different approaches by looking at this problem from different angles; the works are listed below:

- **Phone mapping for multilingual DNN training**

  for multilingual DNN training, using a universal output layer obtained from phone merging can be beneficial if the merged phones are similar and/or some phones are very scarce. In the setting that Flemish plays the role of high resource language and 1 hour of Afrikaans is used as the target language data, employing phone mapping is sensible. We investigated two types of phone mappings; one is based on the knowledge of the native speakers and the other one is a data driven method. The data driven method employs KLD to measure the similarity between the phones. The experimental results revealed that the phone mappings improve the performance compared to the case that each language has its own output layer. Among the phone mapping techniques, we observed that the data driven one slightly outperforms the knowledge-based one.

- **Manifold learning for monolingual systems**

In this thesis, we propose to employ manifold learning to find a feature space with connections to the articulatory features. These features characterize the speech sounds based on the configuration of the speech production system. These features are more compact and few parameters are required to model them. Accordingly, we do not need to have a lot of data to reliably learn a lot of parameters. In our work, we utilized Intrinsic Spectral Analysis (ISA) as a manifold learning technique. We showed that, for GMM-based acoustic modeling in very low resource settings, ISA outperforms the conventional spectral features. However, using DNN-based systems with more data ($>$ 5hr), ISA does not bring consistent improvements. The experiments were carried out on different languages from the GlobalPhone data set as well as Afrikaans from the NCHLT.

- **Manifold learning for crosslingual and multilingual systems**

  Another important contribution of this thesis is the exploitation of manifold learning to obtain a less language dependent features space. This work is motivated by the fact that the articulatory-like and similarly phonological features make a more universal feature space which is a better space in which to transfer knowledge across languages. Our work is based on ISA given the correlation of several of its coordinates to individual articulatory features. We examined training crosslingual and multilingual DNNs on the ISA features and we gained better results compared to the FBANK features. Furthermore, we showed that bottleneck features from DNNs trained on ISA exhibit a high degree of node specificity for phonetic features and that these features are language independent.

- **LRF of multilingual DNNs to improve adaptation**

  In this part of our work, we aim at improving on multilingual DNN adaptation towards low resource target languages. Multilingual DNNs are usually huge in size and parametric and therefore adaptation to low resource target languages with relatively small amounts of data may not succeed. To overcome this problem, we propose to reduce the size of the multilingual DNN. To compress the multilingual DNN, we utilized Low Rank Factorization (LRF) by factorizing the weight layers. This model size reduction, however, usually hurts the performance of the original multilingual DNN; the loss in the performance can be to a large extent recovered by a retraining process. We also observed that even if the compressed network is not as good as the intact one, it is a better starting point for adaptation and ultimately a better performance is obtained. We conducted several experiments by applying LRF on the final weight layer as well as all layers in multilingual DNNs. The results confirm the usefulness of our proposed framework in all scenarios.

- **Sequential LRF for multilingual DNNs**

  Despite the gains obtained from LRF of multilingual DNNs, there still remain some issues in terms of efficiency and accuracy. LRF can be viewed as a regularization where by compressing each layer some noise is added to the network; factorizing all layers equates to adding a huge amount of noise which not only is not a good regularization but also moves the network too far away from its optimal state. Our proposed approach to alleviate this problem is compellingly simple: we deploy the LRF in a sequential manner. In other words, the SVD is applied layer by layer and after each layer is factorized, the DNN is retrained briefly. We showed in experiments that with the sequential LRF we do not need a long recovery phase after the compression and also better results are obtained.

- **Weighted model averaging in distributed multilingual DNNs**

  As another work to improve on multilingual DNNs, we investigated to exploit the similarities between source and target languages. While the consensus in multilingual DNN training is to employ as much data as possible, using only data from similar languages has shown to be effective. Our methodology aims at taking the language similarities into account while no data selection is required. We propose to utilize the distributed DNN training framework for multilingual DNNs so that during the course of training each GPU is used to train a DNN for one language and these DNNs are repeatedly combined with weighted averaging of their parameters. Hence, the impact of each language on the final multilingual DNN can be controlled by the combination weight assigned to its corresponding DNN. Furthermore, we employed this framework as a pre-adaptation where the conventional multilingual DNN is retrained only for a small number of epochs with this methodology so that we do not need to train a multilingual DNN from scratch for any new target language.

- **Cross entropy training of DNN ensemble**

  The final contribution of this thesis is to develop a framework to improve the performance of multilingual DNNs in an ensemble of DNN acoustic models. In this methodology, we would like to exploit two types of information: the first one is the general information which can be obtained from a vast amount of data taken from many different languages; the second type of information is the relevant one taken from individual source languages. Towards this goal, we construct an ensemble of DNNs which includes one generalist and many specialists. The generalist is derived from the conventional multilingual DNN and the specialists are derived from the individual source languages. Then, the constituent models are combined

via a learning algorithm which aims to exploit complementary information in the ensemble. The experiments showed that the performance is always improved and the improvement is more pronounced in under-resourced settings.

8.2 Suggestions for Future Research

In this section we suggest some extensions of the works proposed in this thesis as well as some research directions in line with our works.

- In this thesis we showed that manifold learning can be helpful for low resource ASR specifically in the context of multilingual and crosslingual ASR systems which are the mainstream methodologies in this area. In our work, we employed ISA which involves hyperparameter tuning and has computational costs in large scale speech applications. An interesting alternative is to let the manifold being learned via a DNN; and this can be simultaneously done as an extra task with the common classification tasks or as a regularization in the usual objective function. In the literature, there are a few works addressing manifold learning in the DNN framework ([75, 152]) which can serve as a good starting point to utilize manifold learning in the context of crosslingual and multilingual DNNs.
- In Chapter 6, it is observed that LRF can be successfully employed to improve on multilingual DNN performance. The improvement depends on the compression ratio which is in turn governed by the low rank value. In our work, we did not investigate the best choice of the low rank value. We have found that this value cannot be very small but it would be interesting to explicitly investigate the impact of this value on the multilingual DNN. Moreover, we employed the same low rank value for all layers while we know that different hidden layers do not contain the same amount of information; thus, investigating different compression ratios for different layers could be of value.
- While we have used LRF to compress the multilingual DNNs, there are other possible compression techniques which can play the same role. A popular one is called knowledge distillation; this method has certain advantages compared to the LRF and can be alternatively used for the same purpose.
- In the last chapter, we proposed to benefit from the individual source languages. The weighted model averaging scheme explained in Section 7.2 requires assigning weights to each source language which is done in

a heuristic way in our work. These weights, however, could be also learned together with the DNN parameters and hence we believe it is worth investigating the framework we proposed in Section 7.2 so that the combination weights are learned during the training process.

- This work proposed a novel framework for multilingual DNN training with an ensemble of DNN acoustic models (Section 7.4). In this methodology the linear components of the base models in the ensemble are combined. There are other possible ways to increase the flexibility during the combination; for example, we can let the output of the individual nodes to be combined or we may combine the output of a group of the nodes. This also makes more sense when we look at the outputs of the nodes in hidden layers based on how they respond to the classes of speech sounds [101]

- Finally, it is of interest to investigate to what extent our proposed methods in this thesis are scalable. Although we have presented the performance of our proposed techniques for different scenarios, in all cases the training data size for the target languages was smaller than 30hr. Nowadays, databases including hundreds (and even thousands) of data are commonly used for acoustic modeling; hence, it is valuable to investigate how well the proposed methods in this work can be reused with larger target language data size.

Appendix A

Laplacian Eigenmaps and Intrinsic Spectral Analysis

This appendix details the derivation of the eigendecomposition problems for LE and ISA. As explained in Section 4.3.1, for LE the objective is to minimize the following:

$$\frac{1}{2}\sum_{i,j} w_{ij}\|f(x_i) - f(x_j)\|^2 \tag{A.1}$$

Let's assume $f(.)$ maps the input data to a 1-dimensional space; then we have:

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}}\ \mathbf{f}^T\,\mathbf{L}\mathbf{f} \tag{A.2}$$

with the constraint $\mathbf{f}^T\mathbf{D}\mathbf{f} = 1$, we apply the Lagrangian multipliers:

$$\mathcal{T} = \mathbf{f}^T\mathbf{L}\mathbf{f} + \gamma(1 - \mathbf{f}^T\mathbf{D}\mathbf{f}) \tag{A.3}$$

and

$$\begin{cases} \dfrac{\partial \mathcal{T}}{\partial \mathbf{f}} = 2(\mathbf{L}\mathbf{f} - \gamma\mathbf{D}\mathbf{f}) = 0 \\ \dfrac{\partial \mathcal{T}}{\partial \gamma} = (1 - \mathbf{f}^T\mathbf{D}\mathbf{f}) = 0 \end{cases} \tag{A.4}$$

which leads to the following generalized eigendecomposition problem:

$$\mathbf{Lf} = \gamma \mathbf{Df} \tag{A.5}$$

The ISA optimization problem is a regularized version of the LE:

$$f^* = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \left(\|f\|_K^2 + \xi \mathbf{f}^T \mathbf{Lf} \right) \tag{A.6}$$

The regularization term is $\|.\|_K^2$ which is an inner product in the kernel Hilbert space $(\langle ., . \rangle_K)$ such that:

$$\langle K(x_i, .), K(x_j, .) \rangle_K = K(x_i, x_j) \tag{A.7}$$

Thus, for the solution $f^*(x) = \sum_{i=1}^n a_i K(x_i, x)$, the norm is:

$$\begin{aligned} \|f^*\|_K^2 &= \langle f^*, f^* \rangle_K \\ &= \langle \sum_{i=1}^n a_i K(x_i, x), \sum_{j=1}^n a_j K(x_j, x) \rangle \\ &= \sum_{j,i=1}^n a_i a_j K(x_i, x_j) = \mathbf{a}^T \mathbf{Ka} \end{aligned} \tag{A.8}$$

We also constrain the problem with $\|f\|^2 = \mathbf{a}^T \mathbf{K}^2 \mathbf{a}$. Thus, using the lagrangian multipliers we have:

$$\mathcal{T} = \xi \mathbf{a}^T \mathbf{KLKa} + \mathbf{a}^T \mathbf{Ka} + \gamma(1 - \mathbf{a}^T \mathbf{K}^2 \mathbf{a}) \tag{A.9}$$

$$\begin{cases} \dfrac{\partial \mathcal{T}}{\partial \mathbf{a}} = 2(\xi \mathbf{KLKa} + \mathbf{Ka} - \gamma \mathbf{K}^2 \mathbf{a}) = \mathbf{0} \\ \dfrac{\partial \mathcal{T}}{\partial \gamma} = (1 - \mathbf{a}^T \mathbf{K}^2 \mathbf{a}) = \mathbf{0} \end{cases} \tag{A.10}$$

which leads to the following eigendecomposition problem:

$$(\mathbf{I} + \xi \mathbf{LK})\mathbf{a} = \gamma \mathbf{Ka} \tag{A.11}$$

Bibliography

[1] Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., and Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing 22*, 10 (2014), 1533–1545. pages 22

[2] Ba, J., and Caruana, R. Do deep nets really need to be deep? In *Advances in neural information processing systems* (2014), pp. 2654–2662. pages 101

[3] Banko, M., and Brill, E. Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing. In *Proceedings of the first international conference on Human language technology research* (2001), Association for Computational Linguistics, pp. 1–5. pages 2

[4] Barnard, E., Davel, M. H., van Heerden, C., de Wet, F., and Badenhorst, J. The NCHLT speech corpus of the South African languages. In *SLTU* (St Peterburg, Russia, May 2014), pp. 194–200. pages 27

[5] Bazillon, T., Esteve, Y., and Luzzati, D. Manual vs assisted transcription of prepared and spontaneous speech. In *LREC* (2008). pages 4

[6] Belkin, M., and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data represenation. *Neural Computation 16* (2003), 1373–1396. pages 42, 45

[7] Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from examples. *Machine Learning Research 7* (2006), 2399–2434. pages 42, 44, 45

[8] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. *Advances in neural information processing systems 19* (2007), 153–160. pages 19, 21

[9] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006. pages 18

[10] Bourlard, H., Morgan, N., Wooters, C., and Renals, S. CDNN: A context dependent neural network for continuous speech recognition. In *ICASSP* (1992), vol. 2, IEEE, pp. 349–352. pages 19

[11] Bucilua, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 535–541. pages 101

[12] Burget, L., Schwarz, P., Agarwal, M., Akyazi, P., Feng, K., Ghoshal, A., Glembek, O., Goel, N., Karafiát, M., Povey, D., et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models. In *ICASSP* (2010), IEEE, pp. 4334–4337. pages 24

[13] Byrne, W., Beyerlein, P., Huerta, J. M., Khudanpur, S., Marthi, B., Morgan, J., Peterek, N., Picone, J., Vergyri, D., and Wang, W. Towards language independent acoustic modeling. In *ICASSP* (2000), vol. 2, IEEE, pp. II1029–II1032. pages 24

[14] Caruana, R. Multitask learning. In *Learning to learn*. Springer, 1998, pp. 95–133. pages 26

[15] Cayton, L. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep* (2005), 1–17. pages 44

[16] Chan, W., Ke, N. R., and Lane, I. Transferring knowledge from a RNN to a DNN. In *INTERSPEECH* (2015), pp. 3264–3268. pages 101

[17] Chebotar, Y., and Waters, A. Distilling knowledge from ensembles of neural networks for speech recognition. In *INTERSPEECH* (2016), pp. 3439–3443. pages 101

[18] Chen, D., and Mak, B. K.-W. Multitask learning of deep neural networks for low-resource speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing 23*, 7 (2015), 1172–1183. pages 26

[19] Chen, W., Wilson, J., Tyree, S., Weinberger, K. Q., and Chen, Y. Compressing neural networks with the hashing trick. In *ICML* (2015), pp. 2285–2294. pages 76

[20] Chomsky, N., and Halle, M. The sound pattern of english. pages 58

[21] Chuangsuwanich, E. *Multilingual techniques for low resource automatic speech recognition*. PhD thesis, Massachusetts Institute of Technology, 2016. pages 3, 94, 95

[22] Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., and Andrew, N. Deep learning with COTS HPC systems. In *ICML* (2013), pp. 1337–1345. pages 76

[23] Cohen, P., Dharanipragada, S., Gros, J., Monkowski, M., Neti, C., Roukos, S., and Ward, T. Towards a universal speech recognizer for multiple languages. In *ASRU* (1997), IEEE, pp. 591–598. pages 23

[24] Collins, M. D., and Kohli, P. Memory bounded deep convolutional networks. *arXiv preprint arXiv:1412.1442* (2014). pages 76

[25] Cooley, J. W., Lewis, P. A., and Welch, P. D. The fast fourier transform and its applications. *IEEE Transactions on Education 12*, 1 (1969), 27–34. pages 13

[26] Cui, J., Kingsbury, B., Ramabhadran, B., Saon, G., Sercu, T., Audhkhasi, K., Sethy, A., Nussbaum-Thom, M., and Rosenberg, A. Knowledge distillation across ensembles of multilingual models for low-resource languages. In *ICASSP* (2017), IEEE, pp. 4825–4829. pages 101

[27] Dahl, G. E., Yu, D., Deng, L., and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 1 (2012), 30–42. pages 19

[28] Davis, S., and Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing 28*, 4 (1980), 357–366. pages 13

[29] De Brabanter, K., De Brabanter, J., Suykens, J. A., and De Moor, B. Optimized fixed-size kernel models for large data sets. *Computational Statistics & Data Analysis 54*, 6 (2010), 1484–1504. pages 48

[30] de Wet, F., Kleynhans, N., Van Compernolle, D., and Sahraeian, R. Speech recognition for under-resourced languages: Data sharing in hidden markov model systems. *South African Journal of Science 113*, 1/2 (2017), 1–9. pages 24

[31] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al. Large scale distributed deep networks. In *Advances in neural information processing systems* (2012), pp. 1223–1231. pages 95

[32] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38. pages 18

[33] Demuynck, K. *Extracting, Modelling and Combining Information in Speech Recognition*. PhD thesis, K.U.Leuven, ESAT, February 2001. pages 14

[34] Denil, M., Shakibi, B., Dinh, L., de Freitas, N., et al. Predicting parameters in deep learning. In *NIPS* (2013), pp. 2148–2156. pages 76

[35] D'hoore, B., and Van Compernolle, D. Language independent speech recognition, 2000. U.S Patent 6,085,160. pages 23

[36] Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (2000), Springer, pp. 1–15. pages 101

[37] Drineas, P., and Mahoney, M. W. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research 6* (2005), 2153–2175. pages 46

[38] Duin, R., and Loog, M. Linear dimensionality reduction via a heteroscedastic extension of lda: the chernoff criterion. *IEEE transactions on pattern analysis and machine intelligence 26*, 6 (2004), 732–739. pages 14

[39] Egorova, E., Vesely, K., Karafiát, M., Janda, M., and Cernocky, J. Manual and semi-automatic approaches to building a multilingual phoneme set. In *ICASSP* (2013), IEEE, pp. 7324–7328. pages 25, 32

[40] Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S., and Vincent, P. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Artificial Intelligence and Statistics* (2009), pp. 153–160. pages 21

[41] Fant, G. *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations.* Walter de Gruyter, 1971. pages 42, 43

[42] Furui, S. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing 34*, 1 (1986), 52–59. pages 14, 15

[43] Gales, M. J. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language 12*, 2 (1998), 75–98. pages 22

[44] Gales, M. J. Semi-tied covariance matrices for hidden markov models. *IEEE Transactions on Speech and Audio Processing, 7*, 3 (1999), 272–281. pages 53

[45] Geras, K. J., Mohamed, A.-r., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., and Sutton, C. Blending LSTMs into CNNs. *ICLR* (2015). pages 101

[46] Ghahremani, P., Manohar, V., Povey, D., and Khudanpur, S. Acoustic modelling from the signal domain using CNNs. *INTERSPEECH* (2016), 3434–3438. pages 14

[47] Ghoshal, A., Swietojanski, P., and Renals, S. Multilingual training of deep neural networks. In *ICASSP* (2013), IEEE, pp. 7319–7323. pages 24

[48] Girolami, M. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation 14*, 3 (2002), 669–688. pages 47

[49] Glorot, X., and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Aistats* (2010), vol. 9, pp. 249–256. pages 19

[50] Gokcen, S., and Gokcen, J. A multilingual phoneme and model set: toward a universal base for automatic speech recognition. In *Proc. ASRU* (1997), IEEE, pp. 599–605. pages 23

[51] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. In *ICML* (2013), pp. 1319–1327. pages 22

[52] Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *ICASSP* (2013), IEEE, pp. 6645–6649. pages 22

[53] Grézl, F., Egorova, E., and Karafiát, M. Study of large data resources for multilingual training and system porting. *Procedia Computer Science 81* (2016), 15–22. pages 94

[54] Grezl, F., Karafiát, M., and Janda, M. Study of probabilistic and bottle-neck features in multilingual environment. In *Proc. ASRU* (2011), IEEE, pp. 359–364. pages 32

[55] Grézl, F., Karafiát, M., and Vesely, K. Adaptation of multilingual stacked bottle-neck neural network structure for new language. In *ICASSP* (2014), IEEE, pp. 7654–7658. pages 25, 76

[56] Heeringa, W., and De Wet, F. The origin of afrikaans pronunciation: a comparison to west germanic languages and dutch dialects. In *Proc. of the Conf. of the Pattern Recognition Association of South Africa* (2008), pp. 159–164. pages 32

[57] Heigold, G., Vanhoucke, V., Senior, A., Nguyen, P., Ranzato, M., Devin, M., and Dean, J. Multilingual acoustic models using distributed deep neural networks. In *ICASSP* (2013), IEEE, pp. 8619–8623. pages 95

[58] Hermansky, H. Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America 87*, 4 (1990), 1738–1752. pages 13

[59] Hermansky, H., Ellis, D. P., and Sharma, S. Tandem connectionist feature extraction for conventional HMM systems. In *ICASSP* (2000), vol. 3, IEEE, pp. 1635–1638. pages 24

[60] Hershey, J. R., and Olsen, P. A. Approximating the kullback leibler divergence between Gaussian mixture models. In *ICASSP* (2007), pp. 317–320. pages 35

[61] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine 29*, 6 (2012), 82–97. pages 19, 21

[62] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015). pages 101, 103

[63] Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation 18*, 7 (2006), 1527–1554. pages 19, 21

[64] Hiroya, S., and Honda, M. Estimation of articulatory movements from speech acoustics using an HMM-based speech production model. *IEEE Transactions on Speech and Audio Processing 12*, 2 (2004), 175–185. pages 58

[65] Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *ICASSP* (2013), IEEE, pp. 7304–7308. pages 24, 26, 66, 69

[66] Huang, P.-S., Avron, H., Sainath, T. N., Sindhwani, V., and Ramabhadran, B. Kernel methods match deep neural networks on timit. In *ICASSP* (2014), IEEE, pp. 205–209. pages 46

[67] Huang, X., Baker, J., and Reddy, R. A historical perspective of speech recognition. *Communications of the ACM 57*, 1 (2014), 94–103. pages xvii, 2, 3

[68] Imseng, D., Bourlard, H., and Garner, P. N. Using KL-divergence and multilingual information to improve ASR for under-resourced languages. In *ICASSP* (2012), IEEE, pp. 4869–4872. pages 24

[69] IPA. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999. pages 33

[70] Jakobson, R., Fant, G., and Halle, M. Preliminaries to speech analysis. the distinctive features and their correlates. pages 58

[71] Jansen, A., and Niyogi, P. A geometric perspective on speech sounds. *University of Chicago, Tech. Rep* (2005). pages 43

[72] Jansen, A., and Niyogi, P. Intrinsic Fourier analysis on the manifold of speech sounds. In *ICASSP* (2006), IEEE, pp. 241–244. pages 42, 43, 45, 46

[73] Jansen, A., and Niyogi, P. Semi-supervised learning of speech sounds. In *INTERSPEECH* (2007), pp. 86–89. pages 42

[74] Jansen, A., and Niyogi, P. Intrinsic spectral analysis. *IEEE Transactions on Signal Processing, 61*, 7 (2013), 1698–1710. pages 42, 43, 46, 58, 64, 65

[75] Jansen, A., Sell, G., and Lyzinski, V. Scalable out-of-sample extension of graph embeddings using deep neural networks. *Pattern Recognition Letters* (2017). pages 73, 116

[76] Jansen, A., Thomas, S., and Hermansky, H. Intrinsic spectral analysis for zero and high resource speech recognition. In *INTERSPEECH* (2012). pages 42, 46

[77] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* (2016). pages 16

[78] King, S., and Taylor, P. Detection of phonological features in continuous speech using neural networks. *Computer Speech & Language 14*, 4 (2000), 333–353. pages 58

[79] Köhler, J. Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds. In *Proc. ICSLP* (1996), vol. 4, IEEE, pp. 2195–2198. pages 23

[80] Kullback, S., and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics* (1951), 79–86. pages 35

[81] Kumar, S., Mohri, M., and Talwalkar, A. Ensemble Nyström method. In *In proceedings of Advances in Neural Information Processing Systems* (2009), pp. 1060–1068. pages 46

[82] Lal, P., and King, S. Cross-lingual automatic speech recognition using tandem features. *IEEE Transactions on Audio, Speech, and Language Processing 21*, 12 (2013), 2506–2515. pages 24

[83] Lazaridis, A., Himawan, I., Motlicek, P., Mporas, I., and Garner, P. N. Investigating cross-lingual multi-level adaptive networks: The importance of the correlation of source and target languages. In *Proceedings of the International Workshop on Spoken Language Translation* (2016), no. EPFL-CONF-223756. pages 94

[84] Le, V. B., and Besacier, L. First steps in fast acoustic modeling for a new target language: Application to Vietnamese. In *ICASSP* (2005), pp. 821–824. pages 23, 25

[85] Lee, C.-H., and Gauvain, J.-L. Speaker adaptation based on MAP estimation of HMM parameters. In *ICASSP* (1993), vol. 2, IEEE, pp. 558–561. pages 22

[86] Lee, K.-F. Context-independent phonetic hidden markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 38*, 4 (1990), 599–609. pages 15

[87] Lewis, M. P., Simons, G. F., and Fennig, C. D. *Ethnologue: Languages of the world*, vol. 16. SIL international Dallas, TX, 2009. pages 3

[88] Li, J., Zhao, R., Huang, J.-T., and Gong, Y. Learning small-size DNN with output-distribution-based criteria. In *INTERSPEECH* (2014), pp. 1910–1914. pages 101

[89] Lin, H., Deng, L., Yu, D., Gong, Y.-f., Acero, A., and Lee, C.-H. A study on multilingual acoustic modeling for large vocabulary ASR. In *ICASSP* (2009), IEEE, pp. 4333–4336. pages 24

[90] Liporace, L. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Transactions on Information Theory 28*, 5 (1982), 729–734. pages 18

[91] Liu, W. K., and Fung, P. System and methods for accent classification and adaptation, May 15 2001. US Patent App. 09/858,334. pages 22

[92] Lu, L. *Subspace Gaussian mixture models for automatic speech recognition*. PhD thesis, The University of Edinburgh, 2013. pages 19

[93] Lu, L., Ghoshal, A., and Renals, S. Cross-lingual subspace Gaussian mixture models for low-resource speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing 22*, 1 (2014), 17–27. pages 24

[94] Mall, R., Langone, R., and Suykens, J. A. Furs: Fast and unique representative subset selection retaining large-scale community structure. *Social Network Analysis and Mining 3*, 4 (2013), 1075–1095. pages 46

[95] Mall, R., Langone, R., and Suykens, J. A. Kernel spectral clustering for big data networks. *Entropy 15*, 5 (2013), 1567–1586. pages 46

[96] Miao, Y., Zhang, H., and Metze, F. Distributed learning of multilingual DNN feature extractors using GPUs. In *INTERSPEECH* (2014), pp. 830–834. pages 95

[97] Mohamed, A.-r., Hinton, G., and Penn, G. Understanding how deep belief networks perform acoustic modelling. In *ICASSP* (2012), IEEE, pp. 4273–4276. pages 20, 58

[98] Mohri, M., Pereira, F., and Riley, M. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584. pages 16

[99] Muller, M., Stuker, S., Sheikh, Z., Mtze, F., and Waibel, A. Multilingual deep bottle neck features-a study on language selection and training techniques. In *11th International Workshop on Spoken Language Translation* (2014), pp. 257–264. pages 94, 95

[100] Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis 21*, 1 (2006), 113–127. pages 42

[101] Nagamine, T., Seltzer, M. L., and Mesgarani, N. Exploring how deep neural networks form phonemic categories. In *INTERSPEECH* (2015), pp. 1912–1916. pages 58, 66, 67, 102, 117

[102] Ni, C., Leung, C.-C., Wang, L., Chen, N. F., and Ma, B. Efficient methods to train multilingual bottleneck feature extractors for low resource keyword search. In *ICASSP* (2017), IEEE, pp. 5650–5654. pages 94

[103] Ni, C., Wang, L., Leung, C.-C., Rao, F., Lu, L., Ma, B., and Li, H. Rapid update of multilingual deep neural network for low-resource keyword search. In *INTERSPEECH* (2016), pp. 3698–3702. pages 94, 95

[104] Oostdijk, N. The spoken Dutch corpus. overview and first evaluation. In *International Conference on Language Resources and Evaluation* (2000), pp. 887–894. pages 27

[105] Pan, S. J., and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*, 10 (2010), 1345–1359. pages 5

[106] Panchapagesan, S., and Alwan, A. A study of acoustic-to-articulatory inversion of speech by analysis-by-synthesis using chain matrices and the maeda articulatory model. *The Journal of the Acoustical Society of America 129*, 4 (2011), 2144–2162. pages 58

[107] Papcun, G., Hochberg, J., Thomas, T. R., Laroche, F., Zacks, J., and Levy, S. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. *The Journal of the Acoustical Society of America 92*, 2 (1992), 688–700. pages 58

[108] Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics* (1962), 1065–1076. pages 47

[109] Povey, D., Burget, L., Agarwal, M., Akyazi, P., Feng, K., Ghoshal, A., Glembek, O., Goel, N. K., Karafiát, M., Rastrow,

A., and et al. Subspace Gaussian mixture models for speech recognition. In *ICASSP* (2010), pp. 4330–4333. pages 18

[110] Povey, D., Burget, L., Agarwal, M., Akyazi, P., Kai, F., Ghoshal, A., Glembek, O., Goel, N., Karafiát, M., Rastrow, A., et al. The subspace gaussian mixture model—a structured model for speech recognition. *Computer Speech & Language 25*, 2 (2011), 404–439. pages 19

[111] Povey, D., et al. The KALDI speech recognition toolkit. In *ASRU* (2011), pp. 1–4. pages 16

[112] Povey, D., Zhang, X., and Khudanpur, S. Parallel training of deep neural networks with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455* (2014). pages 95, 96, 101

[113] Principe, J. C. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010. pages 47

[114] Rabiner, L., and Juang, B.-H. *Fundamentals of speech recognition*, 1st ed. Prentice Hall: NJ, USA, 1993. pages 12

[115] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286. pages 15

[116] Rabiner, L. R., and Schafer, R. W. *Digital processing of speech signals*. Prentice Hall, 1978. pages 13

[117] Rahimi, A., and Recht, B. Random features for large-scale kernel machines. In *In proceedings of Advances in neural information processing systems* (2007), pp. 1177–1184. pages 46

[118] Richmond, K. *Estimating articulatory parameters from the acoustic speech signal*. PhD thesis, University of Edinburgh, 2002. pages 58

[119] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290* (2000), 2323–2326. pages 42

[120] Sahraeian, R., and Van Compernolle, D. A study of supervised intrinsic spectral analysis for TIMIT phone classification. In *ASRU* (2013), IEEE, pp. 256–260. pages 7, 42

[121] Sahraeian, R., and Van Compernolle, D. A study of rank-constrained multilingual DNNs for low-resource ASR. In *ICASSP* (2016), IEEE, pp. 5420–5424. pages 8

[122] Sahraeian, R., and Van Compernolle, D. Using weighted model averaging in distributed multilingual DNNs to improve low resource ASR. *Procedia Computer Science 81* (2016), 152–158. pages 9

[123] Sahraeian, R., and Van Compernolle, D. Crosslingual and multilingual speech recognition based on the speech manifold. *IEEE transactions on acoustics, speech, and signal processing* (2017). pages 8

[124] Sahraeian, R., and Van Compernolle, D. Exploiting sequential low-rank factorization for multilingual DNNs. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on* (2017), IEEE, pp. 5025–5029. pages 8

[125] Sahraeian, R., Van Compernolle, D., and de Wet, F. Under-resourced speech recognition based on the speech manifold. In *INTERSPEECH* (2015), pp. 1255–1259. pages 7

[126] Sahraeian, R., Van Compernolle, D., and de Wet, F. Using generalized maxout networks and phoneme mapping for low resource ASR-a case study on Flemish-Afrikaans. In *Pattern Recognition Association of South Africa* (2015), pp. 112–117. pages 7

[127] Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., and Ramabhadran, B. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *ICASSP* (2013), IEEE, pp. 6655–6659. pages 76

[128] Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. Convolutional, long short-term memory, fully connected deep neural networks. In *ICASSP* (2015), IEEE, pp. 4580–4584. pages 22

[129] Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., and Vinyals, O. Learning the speech front-end with raw waveform CLDNNs. In *INTERSPEECH* (2015), pp. 1–5. pages 14

[130] Sak, H., Senior, A., and Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH* (2014), pp. 338–3342. pages 22

[131] Scanzio, S., Laface, P., Fissore, L., Gemello, R., and Mana, F. On the use of a multilingual neural network front-end. In *INTERSPEECH* (2008), pp. 2711–2714. pages 25, 32

[132] Schultz, T., and Kirchhoff, K. *Multilingual speech processing*. Academic Press, 2006. pages 23

[133] Schultz, T., Vu, N. T., and Schlippe, T. Globalphone: A multilingual text & speech database in 20 languages. In *ICASSP* (2013), IEEE, pp. 8126–8130. pages 28

[134] Schultz, T., and Waibel, A. Fast bootstrapping of LVCSR systems with multilingual phoneme sets. In *Eurospeech* (1997). pages 23

[135] Schultz, T., and Waibel, A. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication 35*, 1 (2001), 31–51. pages 23, 25

[136] Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. On parallelizability of stochastic gradient descent for speech DNNs. In *ICASSP* (2014), IEEE, pp. 235–239. pages 95

[137] Senior, A., Heigold, G., Yang, K., et al. An empirical study of learning rates in deep neural networks for speech recognition. In *ICASSP* (2013), IEEE, pp. 6724–6728. pages 21

[138] Silverman, B. W. *Density estimation for statistics and data analysis*. CRC press, 1986. pages 48

[139] Sim, K. C., and Li, H. Robust phone set mapping using decision tree clustering for cross-lingual phone recognition. In *ICASSP* (2008), IEEE, pp. 4309–4312. pages 23

[140] Siniscalchi, S. M., Lyu, D.-C., Svendsen, T., and Lee, C.-H. Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 3 (2012), 875–887. pages 23, 58

[141] Siniscalchi, S. M., Svendsen, T., and Lee, C.-H. Toward a detector-based universal phone recognizer. In *ICASSP* (2008), IEEE, pp. 4261–4264. pages 58

[142] Soman, K., Loganathan, R., and Ajay, V. *Machine Learning with SVM and other Kernel methods*. PHI Learning Pvt. Ltd., 2009. pages 46

[143] Stevens, K. N. *Acoustic phonetics*. Cambridge, MA, USA: MIT Press, 2000. pages 42

[144] Suykens, J. A., Van Gestel, T., and De Brabanter, J. *Least Squares Support Vector Machines*. World Scientific, 2002. pages 47

[145] Swietojanski, P., and Renals, S. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken Language Technology Workshop (SLT), 2014 IEEE* (2014), IEEE, pp. 171–176. pages 22

[146] Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323. pages 42

[147] Thomas, S., Audhkhasi, K., Cui, J., Kingsbury, B., and Ramabhadran, B. Multilingual data selection for low resource speech recognition. In *INTERSPEECH* (2016), pp. 3853–3857. pages 95

[148] Thomas, S., Ganapathy, S., and Hermansky, H. Cross-lingual and multi-stream posterior features for low resource LVCSR systems. In *INTERSPEECH* (2010), pp. 877–880. pages 24

[149] Thomas, S., Ganapathy, S., and Hermansky, H. Multilingual MLP features for low-resource LVCSR systems. In *ICASSP* (2012), IEEE, pp. 4269–4272. pages 26, 66

[150] Thomas, S., Ganapathy, S., Jansen, A., and Hermansky, H. Data-driven posterior features for low resource speech recognition applications. In *INTERSPEECH* (2012). pages 42

[151] Togneri, R., Alder, M., and Attikiouzel, Y. Dimension and structure of the speech space. *IEE Proceedings I (Communications, Speech and Vision) 139*, 2 (1992), 123–127. pages 43

[152] Tomar, V. S., and Rose, R. C. Manifold regularized deep neural networks. In *INTERSPEECH* (2014), pp. 348–352. pages 73, 116

[153] Tompkins, F., and Wolfe, P. J. Approximate intrinsic Fourier analysis of speech. In *INTERSPEECH* (2009), pp. 120–123. pages 46

[154] Tóth, L., Frankel, J., Gosztolya, G., and King, S. Cross-lingual portability of MLP-based tandem features–a case study for english and hungarian. In *INTERSPEECH* (2008), pp. 2695–2698. pages 24

[155] Tüske, Z., Golik, P., Schlüter, R., and Ney, H. Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *INTERSPEECH* (2014), pp. 890–894. pages 14

[156] Van Compernolle, D. Recognizing speech of goats, wolves, sheep and... non-natives. *Speech Communication 35*, 1 (2001), 71–79. pages 24

[157] Variani, E., Bagby, T., McDermott, E., and Bacchiani, M. End-to-End Training of Acoustic Models for Large Vocabulary Continuous Speech Recognition with TensorFlow. *INTERSPEECH* (2017), 1641–1645. pages 2

[158] Veselỳ, K., Karafiát, M., Grézl, F., Janda, M., and Egorova, E. The language-independent bottleneck features. In *Proc. SLT* (2012), pp. 336–341. pages 26, 32, 33, 58, 66

[159] Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory 13*, 2 (1967), 260–269. pages 16

[160] Von Luxburg, U., Belkin, M., and Bousquet, O. Consistency of spectral clustering. *The Annals of Statistics* (2008), 555–586. pages 45

[161] Vu, N. T., et al. Multilingual deep neural network based acoustic modeling for rapid language adaptation. In *ICASSP* (2014), IEEE, pp. 7639–7643. pages 32

[162] Vu, N. T., Metze, F., and Schultz, T. Multilingual bottle-neck features and its application for under-resourced languages. In *Proc. SLTU* (2012), pp. 90–93. pages 26, 58

[163] Vu, N. T., and Schultz, T. Multilingual multilayer perceptron for rapid language adaptation between and across language families. In *INTERSPEECH* (2013), pp. 515–519. pages 94, 95

[164] Ward, T., Roukos, S., Neti, C., Gros, J., Epstein, M., and Dharanipragada, S. Towards speech understanding across multiple languages. In *Fifth International Conference on Spoken Language Processing* (1998). pages 23

[165] Wellekens, C. Explicit time correlation in hidden markov models for speech recognition. In *ICASSP* (1987), vol. 12, IEEE, pp. 384–386. pages 14

[166] Weng, F., Bratt, H., Neumeyer, L., and Stolcke, A. A study of multilingual speech recognition. In *Fifth European Conference on Speech Communication and Technology* (1997). pages 23

[167] Wright, S., and Nocedal, J. Numerical optimization. *Springer Science 35* (1999), 67–68. pages 106

[168] Xu, H., Povey, D., Mangu, L., and Zhu, J. Minimum bayes risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language 25*, 4 (2011), 802–828. pages 111

[169] Xue, J., Li, J., and Gong, Y. Restructuring of deep neural network acoustic models with singular value decomposition. In *INTERSPEECH* (2013), pp. 2365–2369. pages 76, 78, 80

[170] Xue, J., Li, J., Yu, D., Seltzer, M., and Gong, Y. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network. In *ICASSP* (2014), IEEE, pp. 6359–6363. pages 22

[171] Yan, Z., Huo, Q., and Xu, J. A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR. In *INTERSPEECH* (2013), pp. 1172–1183. pages 27

[172] Young, S. J., Odell, J. J., and Woodland, P. C. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology* (1994), Association for Computational Linguistics, pp. 307–312. pages 15

[173] Yu, D., Seide, F., Li, G., and Deng, L. Exploiting sparseness in deep neural networks for large vocabulary speech recognition. In *ICASSP* (2012), IEEE, pp. 4409–4412. pages 76

[174] Yu, D., Seltzer, M. L., Li, J., Huang, J.-T., and Seide, F. Feature learning in deep neural networks-studies on speech recognition tasks. *arXiv preprint arXiv:1301.3605* (2013). pages 58

[175] Zhan, P., and Waibel, A. Vocal tract length normalization for large vocabulary continuous speech recognition. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1997. pages 14

[176] Zhang, S., Bao, Y., Zhou, P., Jiang, H., and Dai, L. Improving deep neural networks for LVCSR using dropout and shrinking structure. In *ICASSP* (2014), IEEE, pp. 6849–6853. pages 76, 79

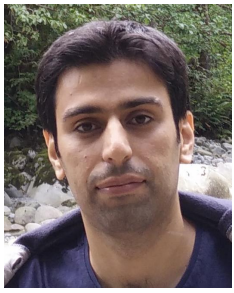
[177] Zhang, X., Trmal, J., Povey, D., and Khudanpur, S. Improving deep neural network acoustic models using generalized maxout networks. In *ICASSP* (2014), IEEE, pp. 215–219. pages 22

[178] Zhang, Y., Chuangsuwanich, E., and Glass, J. Language ID-based training of multilingual stacked bottleneck features. In *INTERSPEECH* (2014), pp. 1–5. pages 94

[179] Zhao, X., and O'Shaughnessy, D. An evaluation of cross-language adaptation and native speech training for rapid HMM construction based on very limited training data. In *INTERSPEECH* (2007), pp. 1433–1436. pages 24

[180] Zhou, G. Adapting to adverse acoustic environment in speech processing using playback training data, July 4 2006. US Patent 7,072,834. pages 22

Short Biography



Reza Sahraeian was born on August 19, 1985 in Shiraz, Iran. He received his B.Sc. and M.Sc. degrees in Electrical Engineering in 2007 and 2010 respectively from Iran University of Science and Technology (IUST), Tehran, Iran. He joined the ESAT-PSI Speech group at KU Leuven in October 2011 as a pre-doctoral student before he started his Ph.D. in September 2012. His research interests include machine learning and its applications for speech recognition specifically low resource and multilingual systems.

List of Publications

Articles in International Journals

1. **Reza Sahraeian**, Dirk Van Compernolle, "*Crosslingual and multilingual speech recognition based on the speech manifold*", Accepted to be published at IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017.

2. Febe de Wet, Neil Kleynhans, Dirk Van Compernolle, **Reza Sahraeian**, "*Speech recognition for under-resourced languages: Data sharing in hidden Markov model systems*", South African Journal of Science, 113 (1/2), 2017.

3. **Reza Sahraeian**, Dirk Van Compernolle, "*Cross-Entropy Training of DNN Ensemble Acoustic Models for Low Resource ASR*", To be submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Articles in International Conferences

1. **Reza Sahraeian**, Dirk Van Compernolle, "*Exploiting sequential low-rank factorization for multilingual DNNs*", In Proc. ICASSP, pages 5025–5029, New Orleans, USA, 5-9 March 2017.

2. **Reza Sahraeian**, Dirk Van Compernolle, "*Using weighted model averaging in distributed multilingual DNNs to improve low resource ASR*", In Proc. SLTU, pages 152–158, Yogyakarta, Indonesia, 9-12 May 2016.

3. **Reza Sahraeian**, Dirk Van Compernolle, "*A study of rank-constrained multilingual DNNS for low-resource ASR*", In Proc. ICASSP, pages 5420–5424, Shanghai, China, 20-25 March 2016.

4. **Reza Sahraeian**, Dirk Van Compernolle, Febe de Wet, "*Using generalized maxout networks and phoneme mapping for low resource ASR - a case study on Flemish-Afrikaans*", In Proc. PRASA-RobMech, pages 112–117, Port Elizabeth, South Africa, 26-27 November 2015.

5. **Reza Sahraeian**, Dirk Van Compernolle, Febe de Wet, " *Under-resourced speech recognition based on the speech manifold*", In Proc. INTERSPEECH, pages 1255–1259, Dresden, Germany, 6-10 September 2015.

6. **Reza Sahraeian**, Dirk Van Compernolle, Febe de Wet, "*On Using Intrinsic Spectral Analysis for Low-resource Languages*", In Proc. SLTU, pages 61–65, Saint-Petersburg, Russia, 14-16 May 2014.

7. **Reza Sahraeian**, Dirk Van Compernolle, "*A study of supervised intrinsic spectral analysis for TIMIT phone classification*", In Proc. ASRU, pages 256–260, Olomouc, Czech Republic, 8-12 December 2013.

Technical Reports

1. **Reza Sahraeian**, Neil Kleynhans, Febe de Wet, Dirk Van Compernolle, "*Knowledge-based phoneme mapping between Flemish and Afrikaans*", Technical report KUL/ESAT/PSI/1502, KU Leuven, ESAT, Leuven, Belgium

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
CENTER FOR PROCESSING SPEECH AND IMAGES (PSI)
Kasteelpark Arenberg 10
B-3001 Heverlee
reza.sahraeian@esat.kuleuven.be
http://www.esat.kuleuven.be